Quantitative Understanding in Biology
Module I: Statistics
Lecture VI: Closing Thoughts and Review

Multiple Hypothesis Testing

So far, we have considered one statistical test at a time. We know that we can control the rate of Type I errors in these tests by setting α appropriately. In many studies several hypothesis will be tested; for example, we may wish to compare means for several groups (related or independent). We may be testing a drug on several different cell lines, or we may be measuring a response at several different time points after application of a treatment. The proper way to analyze data from studies such as these varies depending on the specifics of the experimental design, and we won't be able to go into detail for all them. There is, however, a common theme that needs to be considered: that of multiple hypothesis testing.

The challenge in dealing with multiple hypothesis testing is controlling the Type I error rate across the whole study. If a study involves testing 20 hypotheses, and each has a 5% chance of a Type I error, then the probability of at least one false conclusion is:

```
> 1 - dbinom(0,20,0.05)

[1] 0.6415141
```

In other words, there is a 64% chance that at least one conclusion in our study is wrong. If we want to control the error rate for the study as a whole, we need to adjust α downwards in an appropriate manner. In practice, there are several approaches that are in common use; we'll look at two.

The Bonferroni Correction

The simplest and most conservative approach to controlling a study-wide error rate is the Bonferroni correction. Given a desired study-wide error rate, α, you compute a per-test cutoff, α^*, as follows…

$$\alpha^* = \frac{\alpha}{N}$$

…where N is the number of hypothesis tests in your study. In our example above, we compute α^* = 0.0025. Using this new-per test cutoff, we can estimate the probability of one or more false conclusions…

```
> 1 - dbinom(0,20,0.05/20)
[1] 0.04883012
```


…which is quite close to what we wanted.

In practice, one usually doesn't adjust α, but rather we 'correct' the p-value. For the Bonferroni correction, we multiply the raw p-value by N to compute the corrected p-value, and compare that to our desired study-wide α of 0.05. While this is a little easier in terms of book-keeping, it should be kept in mind that a 'corrected' p-value is not a probability of any particular scenario we've tested. In fact, corrected p-values can be larger than unity (although they are usually reported as 1 in this case).

The Bonferroni correction is the most conservative correction used in multiple hypothesis testing. When the number of hypotheses is small, this is probably an appropriate correction to use.

One of the difficulties in critically evaluating scientific literature is that publications are biased toward reporting statistically significant results. When you see a paper that reports a p-value of 0.02 for a particular test, you have no way of knowing how many other hypotheses have been tested and not reported by the authors.

There can also be a problem when you are 'just looking' at some data you have collected to decide how to analyze it. You are implicitly performing many tests on the data, and selecting only those for which the numbers look hopeful to compute a p-value for. In principle, you should be using some kind of multiple hypothesis control in this case.

Ideally, you would design your experiments and your analyses before collecting any data, and all results, statistically significant or not, would be published. As this is not likely or practical in today's scientific and publishing landscape, it is important to recognize that reported results are probably less certain than they might appear.

The Benjamini-Hochberg Correction for Controlling False Discovery Rate

The advent of high-throughput biological techniques has resulted in a renewed interest in multiple hypothesis testing. New techniques such as microarray and high-throughput sequencing experiments allow for many thousands of data points to be collected in a single experimental protocol; as a result, it is not uncommon to test tens of thousands of hypotheses in a single analysis. In such cases, many practitioners find that the Bonferroni correction is too conservative.

The most common alternative in use today (at least for microarray experiments) is the Benjamini-Hochberg Correction. It works as follows:

1. Order all p-values from smallest to largest, and assign a rank to each one.
2. Correct each p-value by multiplying it by N/rank. This leaves the largest p-value uncorrected, and the smallest with the same adjusted p-value as would have been obtained using the Bonferroni correction.
3. Compare the corrected p-values to your pre-determined α, as usual.

Note that the Benjamini-Hochberg correction seeks to control the "False Discovery Rate", not the "Family-wise Error Rate". This means that we expect some fraction of the significant result to contain

false positives. It is the least conservative correction for multiple hypothesis testing in common use today (short of no correction at all). As such, it is usually applicable to screening studies, where we are trying to identify an enriched set of target genes or compounds for further study, and is usually not the basis on which final scientific conclusions are based.

Additional Comments on Multiple Hypothesis Testing

The Bonferroni and Benjamini-Hochberg p-value corrections (as well as some others) are available in R. The relevant function is `p.adjust`. You need to provide a `method` argument indication which correction you want to use; be sure to specify `method="BH"` (and not `method="hochberg"`) for the Benjamini-Hochberg correction discussed here. Further details are available in the help page for the `p.adjust` method.

There are other techniques for multiple hypothesis testing that are not covered in this course, but are important. In cases where you are comparing means across a two groups (say you are comparing the heights of adult men and women), you would use a t-test. If you are comparing means across several groups (e.g., heights of adults by ethnicity), you could compare each pair of groups using a t-test and then apply a Bonferroni test to account for the multiple hypotheses inherent in the analysis. However, in this case, it is more appropriate to perform an ANOVA test. This would look at all of the data from all of the groups, and tell you if there is any statistically significant difference among any of the groups. If so, then so-called post tests are run to determine which groups differ from each other. ANOVA is a complex topic that has entire texts and courses devoted to it. If you are performing such experiments, you'll need to learn this technique. The relevant function in R are `aov` (or `lm`) for the ANOVA, and `pairwise.t.test` from the post-tests. A good introduction to the theory of ANOVA and its practical implementation using R can be found in Chapter 11 of Michael Crawley's "The R Book" (Wiley, 2007).

Another scenario where a different analysis may be called for is when your data is across multiple groups that have a natural, quantitative ordering. For example, you may be looking at height across different age groups (20-30 years old, 30-39, 40-49, 50-59, etc). In this case, comparison of pairs of groups may not show significant differences, but an analysis that looks for trend might.

Bayesian Statistics

Bayesian statistics is another important topic that we did not cover in this section of the course. Bayesian statistics uses the idea of prior knowledge to compute probabilities. For example, to compute the absolute probabilities of Type I and Type II errors, you need to know roughly how many of the hypotheses you test will be true or not. This is especially important in medical diagnosis and genetic counseling, where prior probabilities can drastically affect computed odds. For example, many couples of Ashkenazi Jewish heritage screen for Tay-Sachs disease before getting married because the prior probability of being a carrier is much higher in the Ashkenazi Jewish population.

The Randomization Test: An Example of Testing By Simulation

In a previous session, we saw how to compare two means using the t-test. This test is based on a model in which the data from the two populations are normally distributed and have the same SD. An alternative method for comparing two means, which does not make these assumptions, is called the randomization test. In practice, the randomization test is used rarely, if ever. However, it is interesting because it works without the need for any complex modeling or assumptions. Additionally, the method forms the basis of a non-parametric test for comparing two means, which we will cover shortly.

We begin with two sets of observations and their means:

$x_1, x_2, x_3, x_4, x_5 \qquad \bar{x}$

$y_1, y_2, y_3, y_4, y_5 \qquad \bar{y}$

The difference between the two observed means is

$$\Delta = \bar{y} - \bar{x}$$

As with the t-test, we wish to ascertain whether this observed difference is statistically significant, or could be due to chance. Our null hypothesis is that the two sets of observations are samples from the same distribution. Interestingly, for the randomization test we do not need to assume anything about this hypothesized distribution.

Now, if the null hypothesis were true, then any of the values we observed would be just as likely to appear in the first set as in the second. In other words, any rearrangement or shuffling of the values we observed (keeping the count of values in each group the same) is just as likely to have been observed as the arrangement we did in fact observe. We can therefore enumerate every possible rearrangement of the values we observed, and compute a Δ for each one. We now have a histogram of Δs that can serve as an estimate for the probability distribution function of Δ. Using this approximate distribution, we can compute the probability of observing given differences in means from two random samples from our hypothesized distribution. We can then compute what proportion of those Δs is equal to or larger in magnitude than the one we observed. This is the p-value corresponding to our null hypothesis. You can look at the range of Δs, and compute a CI of your choosing.

Of course, this p-value is only an estimate. Interestingly, it is a proportion, so, if you are motivated, you can compute a CI for the p-value using techniques we learned earlier.

As mentioned above, the randomization test is rarely, if ever, used in practice. It involves a good deal of bookkeeping to elencate all of the possible rearrangements of the observed values. For all but the most trivial cases, you would need a computer. Even so, with more than a moderate count of observations in each group, the resultant combinatorial explosion would be beyond the capacity of even the most powerful computers. In such cases, sampling a reasonably large number of rearrangements would allow you to develop an estimate of the distribution of Δ, and would allow you to approximate a p-value. The exhaustive procedure is known as the (ostensibly oxymoronic) exact randomization test!

Again, the randomization test is hardly ever used in practice; most sane people would use the t-test if they were comfortable with its assumptions regarding normality and equivalent SDs. That said, if you are comfortable with the idea behind the randomization test, then you have a good understanding of what a p-value is.

The Wilcoxon Rank-Sum Test

The Wilcoxon Rank-Sum test is similar in spirit to the randomization test, and is in fairly common use. It is a non-parametric test that seeks to answer a similar question to the t-test: we again have two sets of observations, and we wish to ascertain whether they come from the same distribution. We are not comfortable with the assumptions of the t-test, and choose a non-parametric method.

Again, we begin with two sets of observations (same as above). We begin our analysis by ordering the values from smallest to largest, and associating a rank with each observation (in the event of a tie, use the average of the ranks to be assigned). Our order might be:

x_4	y_1	y_3	x_2	y_5	y_4	y_2	x_1	x_5	x_3
1	2	3	4	5	6	7	8	9	10

Now we compute the sum of the ranks for each group…

$$x: 8 + 4 + 10 + 1 + 9 = 32$$
$$y: 2 + 7 + 3 + 6 + 5 = 23$$

…and finally a Δ for the difference between the rank sums:

$$\Delta = 23 - 32 = -9$$

Our reasoning from this point on is analogous to that of the randomization test. If the samples were from the same underlying distribution, then any rearrangement or shuffling of the data would be just as likely as the arrangement we observed. We can therefore develop a distribution of Δs, and estimate a p-value that indicates how likely we are to see a Δ as large in magnitude as the one we actually observed.

Note that we never used the actual observed values, just their order. As you might imagine, the Wilcoxon Rank-Sum test is quite robust to outliers; it doesn't matter if the largest value is 10 or 10,000,000; the result would be exactly the same.

The relevant function in R is `wilcox.test`; see the help for details.

Note that the test outlined above is sometimes referred to as the "Mann-Whitney test". To be very precise, we should call it the "two sample Wilcoxon rank sum test". The 'one sample' version is a non parametric test used for paired data, and is available in R using the same `wilcox.test` function, but with the `paired=TRUE` option.

Publication Quality Figures with R

In addition to being a powerful data analysis platform, R is a great tool for preparing publication quality figures. The golden rule for technical illustration is to avoid using any bitmap formats. Sticking with vector based formats allows you to edit your figures with exquisite precision (Adobe Illustrator is recommended for this; PowerPoint is specifically recommended against), and means that you can shrink or enlarge them without loss of clarity. This last point is especially important when preparing posters; you would otherwise have to save your figures with ridiculously high resolution to avoid seeing pixilation and ugly anti-aliasing effects.

To save a figure in R as a PDF (a handy vector format that nearly everyone can read and is compatible with Adobe Illustrator), you begin by opening a PDF file as a graphics device (the file is created in R's current directory):

```
> pdf(file='x.pdf')
> hist(x, probability=TRUE)
> rug(x)
> dev.off()
```

Next, you issue your normal plotting commands. Instead of appearing on the screen, these commands will be sent to the current graphics device, which is your PDF file. To close the device, you issue the `dev.off()` command. Subsequent plotting commands will appear on the screen as usual.

You can also have multiple graphics devices open at once. This allows you to have, say, two windows on the screen, or a screen window and a PDF file open at the same time. Use the `x11()`, `windows()`, and `quartz()` functions to open new screen graphics devices on Linux, Windows, and Mac, respectively.

Plotting commands always go to the current device, which you can change with the `dev.set` function. For help on this function and its friends, see the R help: `?dev.set`

