What if there are multiple (say k) experiments:

- Following ALP example (Handout #1) – suppose that many labs test the drug’s effect on ALP levels
- Significance test of drug effect on ALP considered in laboratory i, at level $\alpha_i$ [indep.]
- Strategy: Suppose we reject $H_0$ (and claim drug increases ALP) only if at least one lab finds significance
- Then overall Type I error rate = $\alpha = P\{\text{reject } H_0 \mid H_0 \text{ true}\}$

$$= 1 - P\{\text{non-sig. result in Lab 1 and non-sig. in Lab 2 and \ldots in Lab k} \mid H_0 \text{ true}\}$$

$$= 1 - P\{\text{non-sig. in Lab 1} \mid H_0 \text{ true}\} \times \ldots \times P\{\text{non-sig. in Lab k} \mid H_0 \text{ true}\}$$

$$= 1 - (1 - \alpha_1) \times \ldots \times (1 - \alpha_k)$$

$$= 1 - (1 - \alpha)^k \quad \text{if} \quad \alpha_i = \alpha \quad \forall i$$

- If $\alpha_i = \alpha_0 = .05$ for all i, visualize effect of larger k … (in SAS)

```sas
/* Look at overall significance level (alpha) as a function of the number of independent centers (k) each testing at level .05 */
data a1;
  do k = 1 to 20;
    output;
  end;
data a1; set a1;
  alpha = 1-(1-.05)**k;
run;
proc sgplot data=a1;
  series x=k y=alpha / lineattrs=(pattern=solid);
title1 'Effect of Multiple Hypothesis Testing';
xaxis label='Number of Independent Tests of Significance';
yaxis label='Overall Type I Error Rate alpha';
run;
```
/* What about Bonferroni correction? */
data a1; set a1;
  bonf = 1-(1-.05/k)**k;
  adj = .05/k;
proc sgplot data=a1;
  series x=k y=bonf / lineattrs=(pattern=solid);
  series x=k y=adj / lineattrs=(pattern=dash) y2axis;
  xaxis label='Number of Independent Tests of Significance';
  yaxis label='Overall Significance Level alpha';
  y2axis label='Bonferroni Significance Threshold (dashed)';
run;
Various strategies
- In this initial strategy, overall Type I error rate gets large for very many labs (larger k) [so this strategy is bad]
- Alternative strategy: similar, but test each lab at level $\alpha_0/k$ (Bonferroni)
  $\rightarrow$ overall level $\alpha$ will be $\leq \alpha_0$ (i.e., the overall Type I error rate is controlled (or protected)
  $\rightarrow$ but $\alpha_0/k$ much too conservative for large k (a common problem in bioinformatics)

Generalized Multiple Hypothesis Testing:

<table>
<thead>
<tr>
<th>Decision</th>
<th>Fail to Reject Null</th>
<th>Reject Null</th>
<th>Total Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null True</td>
<td>U</td>
<td>V</td>
<td>$m_0$</td>
</tr>
<tr>
<td>Null False</td>
<td>T</td>
<td>S</td>
<td>$k-m_0$</td>
</tr>
<tr>
<td></td>
<td>k-R</td>
<td>R</td>
<td>$k$</td>
</tr>
</tbody>
</table>

- $V = \# \text{Type I errors} = \# \text{of false positives}$
- $T = \# \text{of Type II errors} = \# \text{of false negatives}$
- $U+S = \# \text{of correct decisions}$

Example: Gene expression in Alzheimers
(http://www.stat.usu.edu/jrstevens/biostat/data/Alzheimers.csv)

- Central Dogma of Molecular Biology:
  DNA $\rightarrow$ RNA $\rightarrow$ protein $\rightarrow$ biological action
- Differences in biological action may be due to differences in expression at DNA level (estimate based on mRNA levels)
- Data from Sherzer et al. 2007 PNAS paper
  - 37 patients (18 healthy, 19 with Alzheimers)
  - 1056 genes (and expression measurement on each)
  - Want to find which genes are expressed differently in healthy vs. Alzheimers
  - T-test on each gene $\rightarrow$ 1056 p-values
- What we do with these p-values depends on which error rate we want to control
Some error rates we could control:

- **Per-Comparison Error Rate:** PCER = E[V/k]
  - Each of k tests at level \( \alpha \) guarantees:
    - \( PCER \leq \alpha \) (no adjusting for multiplicity – bad!)
  - With \( k=1056 \) and \( \alpha=.05 \), expect how many Type I errors?
    - \( .05 \times 1056 = 53 \) (V) Type I errors, just by chance (pretty high want to reduce this)

- **Family-Wise error Rate:** FWER = P{V \geq 1}
  - Bonferroni correction (each of k tests at \( \alpha/k \)) guarantees:
    - FWER \leq \alpha
  - Sidak correction (less conservative, based on our previous result):
    - \( P_{r} = \min(\frac{k}{k}, P_{r}) \)
  - Sidak correction (less conservative, based on our previous result):
    - \( \tilde{p}_{r} = 1 - (1 - p_{r})^k \)
      - "raw" p-value for test \( r \) (of k)
      - corrected or adjusted p-value; reject \( H_{0,r} \) if \( \tilde{p}_{r} < \alpha \)

- **False Discovery Rate:** FDR = E[ V/R | R > 0]
  - Of great interest in bioinformatics (esp. when k very large)
  - Beware of FDR cheating: Adding several known ‘easy-to-reject’ nulls to the set of tests will artificially inflate R and reduce V/R.
    - Only test what you don’t know \( (H_0) \)
  - Control the FDR with the Benjamini-Hochberg correction:
    - Order p-values \( p_{(1)} \leq p_{(2)} \leq \ldots \leq p_{(k)} \)
    - Define corrected (or adjusted p-values)
      - \( \tilde{p}_{(k)} = p_{(k)} \), and \( \tilde{p}_{(k-i)} = \min \{ \tilde{p}_{(k-i+1)}, \ldots, \tilde{p}_{(k)} \} \)
    - Reject all nulls \( H_{0,r} \) where
      - \( \tilde{p}_{(i)} < \alpha \)
    - This ensures
      - \( FDR \leq \alpha \)
      - FDR controlled at level \( \alpha \)
The BH-correction is an example of a sequential or step-down approach. These tend to be more powerful (smaller p-values) than other approaches because they adjust the i-th most extreme p-value by only k-i+1 tests.

An FDR extension: the q-value

- p-value for a test:
  \[ P \{ T > t \mid H_0 \text{ true} \} \]
  \[ \text{for observed test statistic} \]
  \[ \text{for possible test statistic value} \]
  \[ \text{(from another sample)} \]

- q-value for a test:
  \[ P \{ H_0 \text{ true} \mid T > t \} \]
  \[ \text{for test statistic for a given test} \]
  \[ \text{for other tests} \]
  \[ \text{requires software} \]

= expected proportion of Type I errors incurred when a given test is called significant

Measures significance in terms of the FDR

In general, q-value < BH-corrected p-value

P-values are based on idea of resampling:

Usually rely on statistical theory to look at distribution of all possible test statistics for a given test (each from a different sample of population). But sometimes we actually generate that distribution by resampling from our data (and calculate test statistic for each resampled set).
Can use resampling ideas to adjust for multiple tests:
- Resample (bootstrap or permutations) observations (w/ or w/o replacement), and define
- For each resample, compute p-values for all tests

\[ \tilde{p}_r = \text{proportion of resamples where min. p-value (over family) } \leq p_r \]

- less conservative, incorporates correlations among observations and among tests
- However: these only work well for univariate testing (like each of many labs tests a single outcome)
- multivariate testing (like many genes in a single lab) suffers from hidden assumptions in resampling (actually tests equality of joint distn’s between sample types, across genes) – so be careful with resampling approaches [MCP 2011]

Composite Nulls (special case of Multiple Hypothesis Testing)
- What if want a consensus in a family of tests?
- Suppose \( p_r \) is p-value for null \( H_r \), and \( H_0 = \bigwedge_{r=1}^{k} H_r \); how to get \( p_0 \)?
- One way (quite old, from Fisher), with indep.:

\[ -2 \sum_{r=1}^{k} \log p_r \sim \chi^2_{2k} \]

- Use this idea to obtain adjusted p-values
- Fisher-C adjustment in proc multtest, based on subsets of tests C:

\[ \tilde{p}_r = \max_{C} \left\{ \text{p-value from } -2 \sum_{i \in C} \log p_i \right\} \]

(Nota: this does not give \( p_0 \) for composite null; it’s just based on the same test statistic construction)

- (We’ll come back to this later – meta-analysis)

\[ \text{combine results from multiple studies} \]
/* Look at hundreds of simultaneous tests */
filename myfile "/home/jrstevens/Biostat/Alzheimers.csv"
lrecl=800;
data alzheimers;
    infile myfile dsd delimiter = ',' firstobs=2 missover;
    input variable $ patient1-patient37 $;
    informat variable $9.;
    informat patient1-patient37 12.;
run;

/* Need to transpose data set */
proc transpose data=alzheimers out=alz;
    id variable;
run;

/* Run two-sample t-tests on each gene, and
adjust p-values; TONS of output; see SAS documentation
for details of adjustment methods */
proc multtest data=alz sidak bonferroni holm hochberg fdr;
    class disease;
    test mean(gene1-gene1056);
    contrast 'AD' -1 1;
    title1 'T-tests on all 1056 genes';
run;
T-tests on all 1056 genes

<table>
<thead>
<tr>
<th>Variable</th>
<th>Contrast</th>
<th>Raw</th>
<th>Bonferroni</th>
<th>Stepdown Bonferroni</th>
<th>Sidak</th>
<th>Hochberg</th>
<th>False Discovery Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>gene1</td>
<td>AD</td>
<td>0.6089</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.9970</td>
<td>0.8404</td>
<td></td>
</tr>
<tr>
<td>gene2</td>
<td>AD</td>
<td>0.7131</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.9970</td>
<td>0.8853</td>
<td></td>
</tr>
<tr>
<td>gene3</td>
<td>AD</td>
<td>0.6718</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.9970</td>
<td>0.8655</td>
<td></td>
</tr>
<tr>
<td>gene4</td>
<td>AD</td>
<td>0.8443</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.9970</td>
<td>0.9411</td>
<td></td>
</tr>
<tr>
<td>gene5</td>
<td>AD</td>
<td>0.4836</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.9970</td>
<td>0.7780</td>
<td></td>
</tr>
<tr>
<td>gene6</td>
<td>AD</td>
<td>0.2590</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.9970</td>
<td>0.5825</td>
<td></td>
</tr>
</tbody>
</table>

/* send names of tables to log window */
ods trace on;
proc multtest data=alz sidak bonferroni holm hochberg fdr;
   class disease;
   test mean(gene1-gene1056);
   contrast 'AD' -1 1;
   title1 'T-tests on all 1056 genes';
run;
ods trace off;

----------

Output Added:
----------
Name: Contrasts
Label: Contrast Coefficients
Template: Stat.Multtest.Contrasts
Path: Multtest.Contrasts
----------

Output Added:
----------
Name: pValues
Label: P-Values
Template: Stat.Multtest.pValues
Path: Multtest.pValues
----------
/* run with output suppressed [that part won’t work in SAS Studio; get “ERROR: Insufficient authorization”], and save table called 'pValues' in data set pvals */
ods html close;
proc multtest data=alz fdr stepboot n=10000 seed=1234;
  class disease;
  test mean(gene1-gene1056);
  contrast 'AD' -1 1;
  ods output pValues=pvals;
run;
ods html;

NOTE: The data set WORK.PVALS has 1056 observations and 6 variables.
NOTE: There were 36 observations read from the data set WORK.ALZ.
NOTE: PROCEDURE MULTTEST used (Total process time):
  real time           7.40 seconds
  cpu time            7.11 seconds

/* Visualize results */
proc sort data=pvals; by Raw;
proc sgplot data=pvals;
  series y=FALSEDiscoveryRate x=Raw / lineattrs=(pattern=solid);
  series y=StepDownBootstrap x=Raw / lineattrs=(pattern=dash) y2axis;
  xaxis label='Raw P-value';
  yaxis label='FDR-adjusted P-value';
  y2axis label='Stepdown Bootstrap-adjusted P-value';
title1 'Visualize Adjustments';
run;
/* Look at potentially significant genes */
proc print data=pvals;
  where FalseDiscoveryRate < .10 | StepDownBootstrap < .10;
  var Variable Raw FalseDiscoveryRate StepDownBootstrap;
  title1 'Potentially Significant Genes';
run;

Potentially Significant Genes

<table>
<thead>
<tr>
<th>Obs</th>
<th>Variable</th>
<th>Raw</th>
<th>FalseDiscoveryRate</th>
<th>StepdownBootstrap</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>gene340</td>
<td>0.0002</td>
<td>0.0887</td>
<td>0.0460</td>
</tr>
<tr>
<td>2</td>
<td>gene699</td>
<td>0.0002</td>
<td>0.0887</td>
<td>0.0565</td>
</tr>
<tr>
<td>3</td>
<td>gene366</td>
<td>0.0003</td>
<td>0.0887</td>
<td>0.0679</td>
</tr>
<tr>
<td>4</td>
<td>gene476</td>
<td>0.0003</td>
<td>0.0887</td>
<td>0.0792</td>
</tr>
<tr>
<td>5</td>
<td>gene785</td>
<td>0.0004</td>
<td>0.0887</td>
<td>0.0921</td>
</tr>
<tr>
<td>6</td>
<td>gene749</td>
<td>0.0005</td>
<td>0.0957</td>
<td>0.1121</td>
</tr>
</tbody>
</table>

/* Get adjusted p-values from raw p-values */
proc sort data=pvals; by Variable;
  /* "Variable" is gene name here */
data pvals; set pvals;
  raw_p = Raw; /* must have named raw_p */
proc multtest pdata=pvals sidak;
  title1 'Adjusted P-values';
run;

Adjusted P-values

<table>
<thead>
<tr>
<th>p-Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>