Applied Statistical Genetics in R

for population-based association studies

PART II: Methods to adjust for multiplicity

Outline

1. Measures of error
   - Family-wise error rate
   - False discovery rate
2. Single-step and step-down adjustments
   - Bonferroni adjustment
   - Tukey and Scheffe tests
   - False discovery rate control
3. Resampling-based methods
   - Free step-down resampling
   - Null unrestricted bootstrap

Family-wise error rate

- Single Test:

Table: Type-1 and type-2 errors in hypothesis testing

		Test	
		Non-significant	Significant
Truth	H_0	True Negative	**Type-1 Error**
	H_A	**Type-2 Error**	True Positive

- If the p-value is less than α (typically 0.05) then we reject the null hypothesis in favor of the alternative.

Definitions:

- **level of a test**: (α) probability of making a type-1 error.
- **p-value**: probability of observing something at least as extreme as we do given that the null is true.

Family-wise error rate

- Single Test:
  - Control type-1 error:

$$\text{type-1 error rate } = Pr(\text{reject } H_0 \mid H_0 \text{ is true}) \leq \alpha$$

- Multiple tests:
  - Suppose we are interested in testing K (independent) null hypotheses given by H_{0k}, $k = 1, \ldots, K$:

$$\begin{aligned} FWEC &= Pr(\text{reject at least one } H_{0k} \mid H_{0k} \text{ is true for all } k) \\ &= 1 - Pr(\text{reject no } H_{0k} \mid H_{0k} \text{ is true for all } k) \\ &\leq 1 - (1-\alpha)^K \end{aligned}$$

Family-wise error rate

- For example:
  - $K = 1 \Rightarrow FWEC \leq 0.05$
  - $K = 2 \Rightarrow FWEC \leq 1 - 0.95^2 = 0.0975$
  - $K = 10 \Rightarrow FWEC \leq 0.401$
- Caution: Inflation of type-1 error rate in multiple testing.

Family-wise error rate

- Multiple tests:

Table: Errors for multiple hypothesis tests

		Test: Non-significant	Test: Significant	
Truth	H_0	U	V	m_0
	H_A	T	S	$m - m_0$
		$m - R$	R	m

- V is the number of type-1 errors.
- T is the number of type-2 errors.
- $FWER = Pr(V \geq 1)$; $FWEC = Pr(V \geq 1 \,|H_0^C$ true)

False discovery rate

Definitions:

- **Weak control**: $FWE \leq \alpha$ under the complete null (FWEC)
- **Strong control**: $FWE \leq \alpha$ under all partial subsets of null hypotheses (FWEP).
- **False discover rate (FDR)**: expected proportion of null hypotheses that are true among those that are declared significant.

- Formally, we write:

$$FDR = E\left(\frac{V}{R}\right)$$

- If all null hypotheses are true, then FDR is equal to the $FWER$, i.e. control of FDR leads to control of the FWER in the weak sense.
- In general $FDR \leq FWER$.

False discovery rate

Table: Errors for multiple hypothesis tests

		Test		
		Non-significant	Significant	
Truth	H_0	U	V	m_0
	H_A	T	S	$m - m_0$
		$m - R$	R	m

- Idea of FDR:
  - As the number of false hypotheses $(m - m_0)$ increases, the number of true positives (S) will also increase.
  - In turn, V/R will be smaller and the difference between FDR and the FWER will be greater.
  - Controlling FDR preferable if we want power to detect associations, making some mistakes is acceptable and number of truly false null hypotheses is large.

Single-step and step-down adjustments

- Single-step and step-down adjustments
  - Bonferroni
  - Tukey and Scheffe
  - FDR control
    - Benjamini and Hochberg
    - q-value

Bonferroni adjustment

- Single level α test:

$$Pr(\text{reject } H_0^i \mid H_0^i \text{ true}) \leq \alpha$$

- Multiple (m) tests: Let V be the number of true nulls that are declared significant, then the probability of incorrectly rejecting at least one null hypothesis is given by (assuming independence):

$$\begin{aligned} FWEC &= Pr(V \geq 1 \mid H_0^C \text{ true}) \\ &\leq 1 - (1-\alpha)^m \end{aligned}$$

Bonferroni adjustment

- Replace α with $\alpha' = \alpha/m$ for each test.
  - For example, if $m = 10$ and $\alpha = 0.05$, then $\alpha' = 0.05/10 = 0.005$ and:

$$\begin{aligned} FWEC &\leq 1 - (1 - 0.005)^{10} \\ &= 1 - 0.951 = 0.049 \end{aligned}$$

- If we control each of m tests at the α/m-level, then our overall FWEC will be controlled at a level equal to α.
- Straightforward, but conservative ($\rightarrow$ low power) in genetics setting since SNPs are often correlated.

Bonferroni adjustment

- **Example 1** (Bonferroni adjustment):
  - Suppose we are interested in testing for associations between mutations in protease region of the HIV genome and difference between indinavir (IDV) and nelfinavir (NFV) fold resistance based on the Virco data.
  - Define genotype and trait data:

```
> attach(virco)
> PrMut <- virco[,23:121]!="-" & virco[,23:121]!="."
> NObs <- dim(virco)[1]
> PrMut.sub <-data.frame(PrMut[,apply(PrMut,2,sum)>NObs*.05])
> Trait <- IDV.Fold - NFV.Fold
```

Bonferroni adjustment

- (example continued)
  - Tests of differences in the trait between individuals with and without a mutation at the corresponding sites are calculated based on the t-test. A vector of sorted p-values corresponding to these tests is then reported:

```
> NSites <- dim(PrMut.sub)[2]
> Pvec <- rep(0,NSites)
> for (i in 1:NSites){
+  Pvec[i] <- t.test(Trait[PrMut.sub[,i]==1],
+ Trait[PrMut.sub[,i]==0],na.rm=T)$"p.value"
+  }
> sort(Pvec)
```

Bonferroni adjustment

- (example continued)
  - Based on this unadjusted analysis and an $\alpha = 0.05$, we would conclude that baseline mutations at each of multiple sites (listed below) are associated with a difference in IDV and NFV fold resistance:

```
 [1] 3.732500e-12 9.782323e-10 1.432468e-06 2.286695e-06 5.749467e-06
 [6] 8.924013e-05 4.171618e-04 9.500604e-04 1.115441e-03 1.219064e-03
[11] 1.489381e-03 2.025621e-03 2.556156e-03 4.198935e-03 7.765537e-03
[16] 1.113762e-02 1.557464e-02 1.574864e-02 2.392427e-02 2.508445e-02
[21] 2.722251e-02 3.441981e-02 5.570492e-02 5.748494e-02 6.375590e-02
[26] 1.089171e-01 1.167541e-01 1.556130e-01 2.540249e-01 2.618606e-01
[31] 2.896151e-01 2.945370e-01 3.257741e-01 3.356589e-01 3.441678e-01
[36] 3.619516e-01 3.761893e-01 4.268153e-01 4.480744e-01 4.906612e-01
[41] 5.311825e-01 5.342250e-01 5.440101e-01 6.677043e-01 6.998280e-01
[46] 8.050362e-01 9.938846e-01
```

```
> names(PrMut.sub)[Pvec < 0.05]
```

```
 [1] "P11" "P14" "P30" "P32" "P33" "P35" "P43" "P46" "P47" "P48" "P54"
[13] "P60" "P61" "P67" "P69" "P76" "P82" "P84" "P85" "P88" "P89"
```

Bonferroni adjustment

- (example continued)
  - Bonferroni adjusted p-values are generated using the `p.adjust()` function as follows:

```
> PvecAdj <- p.adjust(Pvec, method="bonferroni")
> sort(PvecAdj)

 [1] 1.754275e-10 4.597692e-08 6.732600e-05 1.074747e-04 2.702250e-04
 [6] 4.194286e-03 1.960660e-02 4.465284e-02 5.242573e-02 5.729603e-02
[11] 7.000090e-02 9.520419e-02 1.201393e-01 1.973500e-01 3.649803e-01
[16] 5.234681e-01 7.320083e-01 7.401862e-01 1.000000e+00 1.000000e+00
[21] 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
[26] 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
[31] 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
[36] 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
[41] 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
[46] 1.000000e+00 1.000000e+00
```

  - Equivalent to taking the original p-values and multiplying by the number of tests (47 in this case).
  - For example, multiplying the smallest p-value, 3.73×10^{-12} by 47 yields the adjusted p-value of 1.75×10^{-10}.

Bonferroni adjustment

- (example continued)
  - Adjusted values that are greater than 1 are set equal to 1 since a p-value is restricted to the closed set $[0, 1]$.
  - Based on this adjustment, we are only able to reject a subset of the null hypotheses that we rejected previously:

```
> names(PrMut.sub)[PvecAdj < 0.05]
[1] "P11" "P30" "P48" "P55" "P76" "P82" "P88" "P89"
```

Tukey and Scheffe tests

- Tukey's Honestly Significantly Different (HSD) test: control of FWE for comparing all pairwise means.
- **Example 2** (Tukey's single-step method):
  - Consider the association between the SNP labelled `resistin_c180g` and the percent change in the non-dominant muscles strength before and after exercise training, as measured by `NDRM.CH`, using the FAMuSS data.

Tukey and Scheffe tests

- (example continued)
  - Unadjusted linear model analysis yields:

```
> attach(fms)
> Trait <- NDRM.CH
> summary(lm(Trait~resistin_c180g))

Call:
lm(formula = Trait ~ resistin_c180g)
Residuals:
    Min      1Q  Median      3Q     Max
-56.054 -22.754  -6.054  15.346 193.946
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)        56.054      2.004  27.973   <2e-16 ***
resistin_c180gCG   -5.918      2.864  -2.067   0.0392 *
resistin_c180gGG   -4.553      4.356  -1.045   0.2964
---
Signif. codes:  0 Ô***Õ 0.001 Ô**Õ 0.01 Ô*Õ 0.05 Ô.Õ 0.1 Ô Õ 1
Residual standard error: 33.05 on 603 degrees of freedom
  (791 observations deleted due to missingness)
Multiple R-squared: 0.007296, Adjusted R-squared: 0.004003
F-statistic: 2.216 on 2 and 603 DF,  p-value: 0.1100
```

Tukey and Scheffe tests

- (example continued)
  - Unadjusted Wald test comparing the mean percent change in muscle strength of individuals with the CG genotype to those with the homozygous wildtype CC genotype yields a significant p-value of 0.039.
  - Estimated coefficient of -5.92 implies that the mean change in muscle strength for individuals with the CG genotype is lower than the mean change among individuals with the CC genotype.

Tukey and Scheffe tests

- (example continued)
  - Applying the Tukey approach provides us with adjusted p-values:

```
> TukeyHSD(aov(Trait~resistin_c180g))

  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Trait ~ resistin_c180g)

$resistin_c180g
            diff        lwr        upr     p adj
CG-CC -5.917630 -12.645660  0.8103998 0.0977410
GG-CC -4.553042 -14.788156  5.6820721 0.5486531
GG-CG  1.364588  -8.916062 11.6452381 0.9478070
```

  - We again see that the difference in means between the individuals with the CG and CC genotypes is -5.92; however, we are unable to detect a significant difference in means between genotype pairs after applying the Tukey adjustment for multiple comparisons. The corresponding adjusted p-value is 0.098.

Tukey and Scheffe tests

- Scheffe's method for controlling FWER under larger set of hypotheses, including all contrasts of the factor level means.

Definition:

- **contrast**: (in the one-way ANOVA setting) a linear combination of the means such that the coefficients sum to zero. More formally, a contrast is written as the function:

$$l = \sum_{i=1}^{m} \lambda_i \mu_i$$

such that $\sum_{i=1}^{m} \lambda_i = 0$.

Tukey and Scheffe tests

- Scheffe's method involves constructing a slightly modified F-statistic given by:

$$F_s = \frac{\rho' Y/(m-1)}{MSE(\rho'\rho)} \sim F_{m-1,(m\times n)-m}$$

- The adjustment for multiple comparisons enters into the test statistic through the degrees of freedom. In this case, the numerator degrees of freedom are set equal to $m-1$ where in the usual setting for testing a single contrast, we set this equal to 1.

Tukey and Scheffe tests

- **Example 3**:
  - Consider $m = 5$ groups each of size $n = 20$. Suppose an investigator decides to test the null hypothesis $H_0 : \mu_1 = \mu_2$ at the $\alpha = 0.05$ level without making a multiple comparison adjustment and calculates the F-statistic to be 4.5. Comparing it to the critical value given by $F_{1,40-1} = 4.08$, it is concluded that the null hypothesis is false and indeed there is a difference in the two population means.
  - Using Scheffe's method, we would instead get $F_s = 4.5/(5-1) = 1.125$ and compare it to the critical value of $F_{4,120-4} = 2.45$. In this case, we would conclude that based on the observed data, we can not reject the null hypothesis that the two population means are equal.
- While a function for implementing Scheffe's test in R is not readily available, it is straightforward to apply this approach using existing functions and the identify derived above.

False discovery rate control

- Benjamini and Hochberg (B-H) adjustment
  1. Let $p_{(1)}, \ldots, p_{(m)}$ denote the ordered observed p-values such that:

     $$p_{(1)} \leq \cdots \leq p_{(m)}$$

     and let the corresponding null hypotheses be given by $H_0^{(1)}, \ldots, H_0^{(m)}$.
  2. Define:

     $$k = max \left\{ i : p_{(i)} \leq \frac{i}{m} q \right\} \quad (1)$$

     where q is the level at which we want to control the FDR.
  3. Reject $H_0^{(1)}, H_0^{(2)}, \ldots, H_0^{(k)}$.

False discovery rate control

- **Example 4**:
  - Suppose for each of 10 SNPs we construct a 3×2 contingency table and calculate a χ^2-statistic corresponding to the null hypothesis $H_0 : OR_i = 1$.
  - Let the resulting ordered p-values be given by:

$$0.001 \quad 0.012 \quad 0.014 \quad 0.122 \quad 0.245$$
$$0.320 \quad 0.550 \quad 0.776 \quad 0.840 \quad 0.995$$

  - The Bonferroni adjustment would lead to us to use the adjusted significance level of $\alpha^* = 0.05/10 = 0.001$. Based on this, we would reject only $H_0^{(1)}$.

False discovery rate control

- (example continued)
  - Using the B-H method, we would instead compare the *ith* ordered p-value to $\alpha_i^* = 0.05\,(i/10)$, given by:

$$0.005 \quad 0.010 \quad 0.015 \quad 0.020 \quad 0.025$$
$$0.030 \quad 0.035 \quad 0.040 \quad 0.045 \quad 0.050$$

  - The value k is defined as the maximum i such that $p_{(i)}$ is less than or equal to α_i^*.
  - In this example, we have $k = 3$ since $p_{(3)} = 0.014 < 0.015$ while $p_{(i)} > \alpha_i^*$ for $i > 3$. Thus, using the B-H approach we would reject the null hypotheses $H_0^{(1)}$, $H_0^{(2)}$ and $H_0^{(3)}$.
  - Importantly, the fact that $p_{(2)} = 0.012$ is not less than $\alpha_i^* = 0.010$ is not relevant since the larger p-value $p_{(3)} = 0.014$ does meet its rejection criterion.

False discovery rate control

- We call this procedure a *step-down adjustment* since each test statistic has a different criterion for rejection.
- In addition to defining rejection criteria, we can also calculate adjusted p-values.
  - Calculate an adjusted p given by $p^{adj}_{(i)} = p_{(i)}m/i$ for each i.
  - Update these p-values to ensure monotonicity by letting $p^{adj}_{(i)} = min_{j \geq i}\left(p^{adj}_{(j)}\right)$.

False discovery rate control

- **Example 5** (B-H adjusmtent):
  - Returning the Virco example above, recall `Pvec` was vector of unadjusted p-values.

```
> m <- length(Pvec)
> BHp <- sort(Pvec,decreasing=T)*m/seq(m,1)
> BHp[order(Pvec,decreasing=T)] <- cummin(BHp)
> sort(BHp)

 [1] 1.754275e-10 2.298846e-08 2.244200e-05 2.686866e-05 5.404499e-05
 [6] 6.990477e-04 2.800943e-03 5.581605e-03 5.729603e-03 5.729603e-03
...
[41] 5.946157e-01 5.946157e-01 5.946157e-01 7.132296e-01 7.309314e-01
[46] 8.225370e-01 9.938846e-01
```

False discovery rate control

- (example continued)
  - The resulting sites that are declared significant based on the B-H adjustment are given by:

```
> names(PrMut.sub)[BHp < 0.05]

 [1] "P11" "P30" "P43" "P46" "P47" "P48" "P54" "P55" "P60" "P61" "P67
[13] "P76" "P82" "P84" "P85" "P88" "P89"
```

  - Notably this is a subset of the sites found based on an unadjusted analysis and less conservative than the Bonferroni adjustment described in Example above.
  - Finally, the same adjusted p-values can also be calculated by specifying `method="BH"` within the `p.adjust()`function as follows:

```
> sort(p.adjust(Pvec, method="BH"))

 [1] 1.754275e-10 2.298846e-08 2.244200e-05 2.686866e-05 5.404499e-05
 [6] 6.990477e-04 2.800943e-03 5.581605e-03 5.729603e-03 5.729603e-03
...
[41] 5.946157e-01 5.946157e-01 5.946157e-01 7.132296e-01 7.309314e-01
[46] 8.225370e-01 9.938846e-01
```

False discovery rate control

- The B-H procedure for controlling the FDR assumes independence of the test statistics corresponding to the true null hypotheses.
- Benjamini and Yekutieli (B-Y) propose an extension of the B-H approach
- Replace q with the quantity $\tilde{q} = q / \sum_{i=1}^{m} (1/i)$

False discovery rate control

- **Example 6** (B-Y adjustment):
  - Again using the vector of p-values from above examples, the B-Y adjustment is calculated using the `p.adjust()` function as follows:

```
> BYp <- p.adjust(Pvec, method="BY")
> sort(BYp)

 [1] 7.785410e-10 1.020220e-07 9.959678e-05 1.192422e-04 2.398497e-04
 [6] 3.102348e-03 1.243048e-02 2.477096e-02 2.542777e-02 2.542777e-02
...
[41] 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
[46] 1.000000e+00 1.000000e+00
```

  - The resulting *p*-values are more conservative than we saw with the application of the B-H approach.
  - The SNPs corresponding to significant associations are now given by:

```
> names(PrMut.sub)[BYp < 0.05]

 [1] "P11" "P30" "P43" "P48" "P54" "P55" "P60" "P61" "P76" "P82" "P85"
[13] "P89"
```

False discovery rate control

- The *q-value* is an alternative measure of significance that is also based on the FDR concept and was proposed for genomewide studies.
- Similar to a *p*-value, the *q*-value can be thought of as the expected proportion of false positives among all features that are as extreme as or more extreme than the feature under consideration.
- Use of the *q*-value is most appropriate when the number of tests performed is large, in which the probability that at least one test is declared significant is close to one.
- If we set the tuning parameter λ equal to 0, the *q*-value results in the same adjusted p-values as the B-H method described above; however, this is a conservative estimate of the *q*-value.

False discovery rate control

- **Example 7** (Calculation of the q-value):
  - Use qvalue package:

```
> install.packages("qvalue")
> library(qvalue)
> sort(qvalue(Pvec,lambda=0)$qvalues)

 [1] 1.754275e-10 2.298846e-08 2.244200e-05 2.686866e-05 5.404499e-05
 [6] 6.990477e-04 2.800943e-03 5.581605e-03 5.729603e-03 5.729603e-03
...
[41] 5.946157e-01 5.946157e-01 5.946157e-01 7.132296e-01 7.309314e-01
[46] 8.225370e-01 9.938846e-01
```

  - If instead of specifying λ, we specify to use the bootstrap estimation method, `pi0.method="bootstrap"`, we get less conservative estimates of the q-values:

```
> sort(qvalue(Pvec,pi0.method="bootstrap")$qvalues)

 [1] 2.488334e-11 3.260774e-09 3.183262e-06 3.811158e-06 7.665956e-06
 [6] 9.915570e-05 3.972969e-04 7.917170e-04 8.127096e-04 8.127096e-04
...
[41] 8.434265e-02 8.434265e-02 8.434265e-02 1.011673e-01 1.036782e-01
[46] 1.166719e-01 1.409765e-01
```

False discovery rate control

- (example continued)
  - In this case, many more sites appear to be significant predictors of a difference in IDV and NFV fold resistance.
  - The `qvalue()` function can also give us estimate of the proportion of true null hypotheses, given by m_0/m where m_0 and m are respectively the numbers of true null hypotheses and total hypotheses.
  - In this example, this proportion is given by:

```
> qvalue(Pvec,pi0.method="bootstrap")$pi0
[1] 0.1418440
```

Resampling-based methods

- Alternative to the single-step and step-down procedures described above that involve taking repeated samples from the observed data.
- One primary advantage of resampling-based methods is that they offer a natural approach to account for underlying, unknown correlation structure among multiple hypotheses.
- Approaches:
  1. Free step-down resampling (FSDR) method of Westfall and Young (assumes subset pivotality)
  2. Null unrestricted bootstrap of Pollard and van der Laan.

Free step-down resampling

- Idea:
  1. By taking repeated samples of the observed data, we can simulate the distribution of the test statistics (or p-values) under the complete null hypothesis, H_0^C.
  2. Comparing the observed test statistics to this empirical distribution allows us to ascertain the corresponding significance of our tests.

Definition:

- **subset pivotality**: the distribution of test statistics is the same under any combination of true null hypotheses. That is, the test statistic distribution is invariant to whether all null hypotheses are indeed true (H_0^C) or a partial set of null hypotheses are true.
  - Specifically, the covariance between test statistics is assumed to be the same under all scenarios of true and false null hypotheses.
  - Under this assumption, importantly, error control under the complete null will give us the desired control under the true data generating distribution.

Free step-down resampling

FSDR procedure:

1. Determine "observed" test statistics and p-values.
   - Let x_j represent our genotype variables for $j = 1, \ldots, m$ and suppose the phenotype under study is given by y.
   - Construct the following linear model:

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_m x_{im} + \epsilon_i \tag{2}$$

     for $i = 1, \ldots, n$ where n is our sample size and we assume $\epsilon_i \sim N(0, \sigma^2)$.
   - Using ordinary least squares regression, we arrive at an estimate, $\widehat{\beta}$, of the vector of parameters $\beta = (\beta_0, \beta_1, \ldots, \beta_m)'$.

Free step-down resampling

1. (continued)
   - For each β_j, construct a test statistic and p-value, given by T_j and p_j respectively, corresponding to the null hypothesis of $H_0 : \beta_j = 0$. For example, T_j may be a Wald test statistic and $p_j = Pr(|T_j| > t_{(n-1),(1-\alpha)/2})$.
   - Let the corresponding ordered test statistics, sorted from smallest to largest, be given by $T_{(1)}, \ldots, T_{(m)}$.
2. Generate (approximate) distribution of test statistics under the complete null.
   - Determine the residuals from the model fitting procedure described in Step 1, given by $\widehat{r}_i = y_i - x_i^T \widehat{\beta}$, where $x_i^T = (x_{i1}, \ldots, x_{im})$.

Free step-down resampling

2. (continued)
   - Sample with replacement from these residuals to get a bootstrap data set. That is, for $i = 1, \ldots, n$, we let $y_i^* = \widehat{r}_i^*$ where $\widehat{r}_i^*$ is drawn with replacement from the original set of residuals: $\widehat{r}_1, \ldots, \widehat{r}_n$.
   - Using these new data, $y_1^*, \ldots, y_n^*$ as our response and the original design matrix X, refit the model and determine corresponding test statistics. Let these be $T_{(1)}^*, \ldots, T_{(m)}^*$ where the ordering is the same as the ordering of the original test statistics.
   - Note that at this step, the T^* are not necessarily ranked from smallest to largest, or in other words, *monotonicity* does not hold. These resulting test statistics are one realization from the complete null generating distribution.

Free step-down resampling

3. Compare observed test statistics to test statistics under the complete null to get adjusted p-values.
   - Repeat Step 2 above B times to arrive at multiple bootstrap samples.
   - For each sample, successive maxima are defined as follows:

$$
\begin{aligned}
q_1^* &= |T_1|^* \\
q_2^* &= max(q_1^*, |T_2|^*) \\
q_3^* &= max(q_2^*, |T_3|^*) \\
&\vdots \\
q_m^* &= max(q_{(m-1)}^*, |T_m|^*)
\end{aligned}
$$

and we determine whether $|T_{(j)}| > q_j^*$. That is, check whether the *jth* ordered test statistic is greater than the corresponding statistic that was generated based on the distribution of test statistics under the complete null.

Free step-down resampling

3. (continued)
   - The adjusted p-values, given by $\tilde{p}_{(j)}$ for $j = 1, \ldots, m$, are then defined as the proportion of the B bootstrap samples for which this inequality holds. More formally, we write:

$$\tilde{p}_{(j)} = \frac{1}{B} \sum_{b=1}^{B} I\left(T_{(j)} > q_j^{*(b)}\right)$$

   where $I(\cdot)$ is the indicator function which equals 1 if the argument is true and 0 otherwise, and b indicates the specific bootstrap sample.

Free step-down resampling

3. (continued)
   - Finally, monotonicity of these resulting adjusted p-values is ensured by completing this final step:

$$\begin{aligned} \tilde{p}^*_{(m)} &= \tilde{p}^*_{(m)} \\ \tilde{p}^*_{(m-1)} &= max(\tilde{p}^*_{(m)}, \tilde{p}^*_{(m-1)}) \\ &\vdots \\ \tilde{p}^*_{(1)} &= max(\tilde{p}^*_{(2)}, \tilde{p}^*_{(1)}) \end{aligned}$$

Free step-down resampling

- Under subset pivotality, this approach controls FWE in the strong sense.
- Also called the *maxT* procedure.
- Replacing test statistics with p-values and taking successive minimum yields the *minP* approach.
- Example below uses direct coding. Alternatively, the multtest package in R includes mt.maxT() and mt.minP() functions. These were developed fro expression data and application to SNP data is not always straightforward.

Free step-down resampling

- **Example 8** (Free step-down resampling adjustment for a quantitative trait):
  - Returning the FAMuSS data, we consider whether there is an association between change in muscle strength of the non-dominant arm and the presence of two variant alleles for each of the four SNPs within the actn3 gene.
  - Fit multivariable model:

```
> attach(fms)
> Actn3Bin <- data.frame(actn3_r577x!="TT",actn3_rs540874!="AA",
+ actn3_rs1815739!="TT",actn3_1671064!="GG")
> Mod <- summary(lm(NDRM.CH~.,data=Actn3Bin))
> Mod
```

Free step-down resampling

- (example continued)

```
Call:
lm(formula = NDRM.CH ~ ., data = Actn3Bin)

Residuals:
    Min      1Q  Median      3Q     Max
-55.181 -22.614  -7.414  15.486 198.786

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                       54.700      3.212  17.028   <2e-16 ***
actn3_r577x.....TT.TRUE          -12.891      4.596  -2.805   0.0052 **
actn3_rs540874.....AA.TRUE        10.899     11.804   0.923   0.3562
actn3_rs1815739.....TT.TRUE       27.673     17.876   1.548   0.1222
actn3_1671064.....GG.TRUE        -29.166     17.516  -1.665   0.0964 .
---
Signif. codes:  0 Ô***Õ 0.001 Ô**Õ 0.01 Ô*Õ 0.05 Ô.Õ 0.1 Ô Õ 1

Residual standard error: 32.93 on 591 degrees of freedom
  (801 observations deleted due to missingness)
Multiple R-squared: 0.01945, Adjusted R-squared: 0.01281
F-statistic:  2.93 on 4 and 591 DF,  p-value: 0.02037
```

Free step-down resampling

- (example continued)
  - Record "observed" test statistics, given by $-2.81, 0.923, 1.55$ and -1.67:

```
> TestStatObs <- Mod$coefficients[-1,3]
> Tobs <- as.vector(sort(abs(TestStatObs)))
```

  - Before applying the resampling procedure, we need to subset the data that went into the above analysis:

```
> MissDat <- apply(is.na(Actn3Bin),1,sum)>0 | is.na(NDRM.CH)
> Actn3BinC <- Actn3Bin[!MissDat,]
```

  - We also need to record the ordering of our original test statistics:

```
> Ord <- order(abs(TestStatObs))
```

Free step-down resampling

- (example continued)
  - The next step of the FSDR involves re-sampling from the residuals and arriving at test statistics under the null generating distribution:

```
> M <- 1000
> NSnps <- 4
> Nobs <- sum(!MissDat)
> TestStatResamp <- matrix(nrow=M,ncol=NSnps)
> for (i in 1:M){
+ Ynew <- sample(Mod$residuals,size=Nobs,replace=T)
+ ModResamp <- summary(lm(Ynew~.,data=Actn3BinC))
+ TestStatResamp[i,] <- abs(ModResamp$coefficients[-1,3])[Ord]
+ }
```

  - We see that in each iteration of the `for` loop we:
    1. take a sample from the model residuals with replacement
    2. refit the model using this sample as the new outcome and
    3. record the test statistics corresponding to our ordered observed statistics, given by `Tobs`.

Free step-down resampling

- (example continued)
  - The result is a matrix of test statistics, called `TestStatResamp`, corresponding to the null distribution.
  - The final step is to compare our observed test statistics to the the distribution of test statistics we generated:

```
> MaxT <- function(x){
+ xor <- x
+ for (i in 2:(length(x))){
+ xor[i] <- max(xor[i-1],xor[i])
+ }
+ return(xor)
+ }
> Qmat <- t(apply(TestStatResamp,1,MaxT))
```

  - Adjusted p-values are then given by:

```
# Note that your code will result in slightly different
# values since we took a random sample above.
> Padj <- 1-apply(t(matrix(rep(Tobs,M),NSnps)) > Qmat,2,mean)
> Padj

[1] 0.310 0.203 0.203 0.034
```

Free step-down resampling

- (example continued)
  - Monotonicity of the resulting p-values is already achieved, so an additional step is not needed.
  - Based on this analysis we conclude that individuals who are homozygous variant at the `actn_577x` SNP have a significantly lower percent change in the non-dominant arm muscle strength than individuals who are homozygous wildtype or heterozygous at this SNP (adjusted $p = 0.034$).

Null unrestricted bootstrap

- Recall, for the free step-down approach, we resample data from the complete null distribution and then generate the test statistic distribution based on these resampled data.
- Now we will instead arrive at the test statistic distribution based on the original data. In turn, the projection of this distribution onto the space of mean zero distributions yields asymptotic strong control of FWE.

Null unrestricted bootstrap

- **Example 9** (Null unrestricted bootstrap approach)
  - Returning to the data setting and model from Example above, we begin by defining the estimated coefficients based on the model denoted `Mod`:

```
> CoefObs <- as.vector(Mod$coefficients[-1,1])
```

  - The following `for` loop is then applied where `Nobs`, `MissDat` and `Actn3BinC` are defined above:

```
> B <-1000
> TestStatBoot <- matrix(nrow=B,ncol=NSnps)
> for (i in 1:B){
+ SampID <- sample(1:Nobs,size=Nobs, replace=T)
+ Ynew <- NDRM.CH[!MissDat][SampID]
+ Xnew <- Actn3BinC[SampID,]
+ CoefBoot <- summary(lm(Ynew~.,data=Xnew))$coefficients[-1,1]
+ SEBoot <- summary(lm(Ynew~.,data=Xnew))$coefficients[-1,2]
+ if (length(CoefBoot)==length(CoefObs)){
+ TestStatBoot[i,] <- (CoefBoot-CoefObs)/SEBoot
+ }
+ }
```

Null unrestricted bootstrap

- (example continued)
  - We see here that we begin by drawing a bootstrap sample from the data (both the trait and genotypes) without disrupting the within individuals link. We then fit a model based on these data and calculate the vector of test statistics.
  - Finally, we determine a significant threshold as follows:

```
> for (cj in seq(2.7,2.8,.01)){
+ print(cj)
+ print(mean(apply(abs(TestStatBoot)>cj,1,sum)>=1,na.rm=T))
+ }
# Note that, depending on your sample,
# a different range for cj may be required

[1] 2.7
[1] 0.06471183
[1] 2.71
[1] 0.06268959
[1] 2.72
[1] 0.05965622
[1] 2.73
[1] 0.0586451
```

Null unrestricted bootstrap

- (example continued)

```
[1] 2.74
[1] 0.05662285
[1] 2.75
[1] 0.05460061
[1] 2.76
[1] 0.05257836
[1] 2.77
[1] 0.05055612
[1] 2.78
[1] 0.04954499
[1] 2.79
[1] 0.04954499
[1] 2.8
[1] 0.04853387
```

  - A significance threshold of 2.78 for all four tests maintains a type-1 error rate of less than 0.05.
  - Comparing this to the observed test statistics given above, we again conclude that the `actn3_577x` SNP is significantly associated with percent change in the non-dominant arm muscle strength.

Summary

- Topics covered:
  - FWE and FDR
  - Bonferroni adjustment
  - Tukey and Scheffe tests
  - B-H, B-Y and q-value
  - Free step-down (MaxT) and null unrestricted bootstrap
- Useful R packages and functions:
  - qvalue: qvalue()
  - multtest: mt.maxT(); mt.minP()
  - generic: t.test(); sort(); p.adjust(); lm(); TukeyHSD(); sort(); order(); cummin(); max(); apply()