Multiple Testing

False Discovery Rates

BTRY/STSCI 4520

Multiple Testing

Permutation tests widely used in micro-array analysis.

- Typical (early) experiment – measure expression of 100 genes at once.
- Usually on 8 patients. 4 with condition/disease, 4 without.
- For each gene, conduct permutation test for significant difference in expression.

A null data set:

```
nsub = 8
ngene = 100
data = matrix(rnorm(nsub*ngene),nsub,ngene)
label = c(rep(1,4),rep(2,4))
```

A Fishing Expedition

Issues with this approach

- Researchers have no particular knowledge of which genes will be associated with disease.
- Will publish any genes that show up as significant.
- But, with 100 genes, there is much more chance that we'll find something significant, even if none of them play any role in disease.
- How bad is this? We'll try a simulation.

Running 100 Tests

First of all let's look at the standard *t*-test.

```
gene.t.stat = function(matrix){
   t.gene = rep(0,100)
   for (i in 1:100){
      t.gene[i] = abs(t.test(matrix[1:4,i],matrix[5:8,i])$stat)
   }
   return(t.gene)
}
t.obs = gene.t.stat(data)

plot(t.obs)
abline(h = qt(0.975,7))
```

100 Permutation Tests

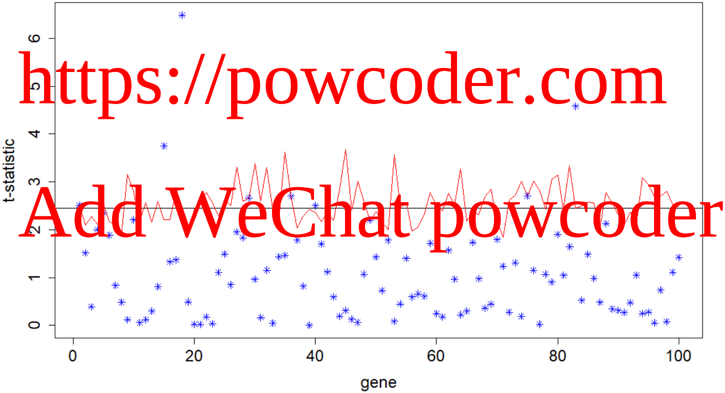
```
nperm = 1000
t.perm = matrix(rep(0,nperm*100),nrow=nperm)
for (i in 1:nperm){
  #Create a new permutation to be used for each iteration
    S = sample(8)
  #Rearrange the row indices of the data each time, according
  #to the permutation
    temp = rbind(data[S[1:4],],data[S[5:8],])
  #Assign the ith row of our permutations by yielding 100
  #permutated t-statistics
    t.perm[i,] = gene.t.stat(temp)
}
```

Look at the quantiles of the permutation distribution and count significant genes

```
t.quantile = apply(t.perm,2,quantile,0.95)
sum( t.obs > t.quantile)
```

Graphically

We reject any test that falls above the standard *t*-cutoff, or above the permutation critical value.



A Correction

- If the most significant gene is significant, we will report it.
- Measure significance by the size of the t statistic.
- We'd like to ensure that the probability of *any* false positive is α.
- Classical approach is Bonferroni correction: require significance at $\alpha/$ngene. Then if p-values are $p_1, \ldots, p_k$

$$
\begin{aligned}
P(\text{any Type I error}) &= P(p_1 < \alpha/k, p_2 < \alpha/k, \ldots p_k < \alpha/k) \\
&\leq P(p_1 < \alpha/k) + \ldots + P(p_k < \alpha_k) \\
&\leq \alpha/k + \ldots \alpha/k \\
&= \alpha
\end{aligned}
$$

because p-values are uniform under the null:

$$P(p_j < \tau) = P(F_T(T_j) < \tau) = P(T_j < F_T^{-1}(\tau)) = \tau$$

When Tests are Comparable

- Bonferroni correction is very conservative (going from “and” to “sum” is pretty strong)

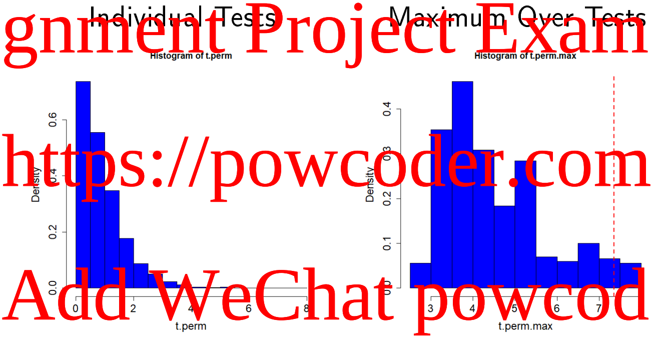
- Instead, we can look at the original: we report a discovery if *the largest* statistic (or smallest p-value) is over a threshold. Let’s just find a good threshold.

- For each permutation measure *maximum* t-statistic over 100 genes and use the distribution of these (over permutations) for a critical value.

```
t.perm.crit = quantile(t.perm.max,.95)
sum(t.obs>t.perm.crit)
```

Ideally, we would run a simulation to perform permutation tests on 1000 data sets to check on these α-levels.

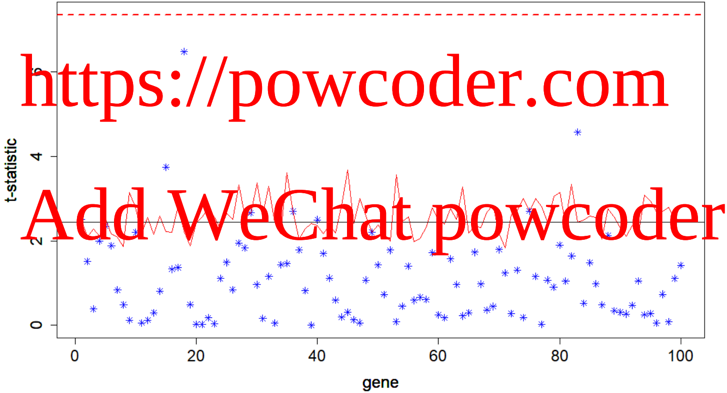
A Comparison of Null Distributions



Maximum t-statistic will correspond to a different gene each distribution, but indicates how bad looking over 100 genes can be.

Controlling Family-Wise Error Rate

The *maximum* critical value is a way to ensure that the probability of even one error is small.



Killing the Power of a Test

By taking a maximum, it's much less likely for a real effect to pass the threshold.

Data where there is a real effect.

```
data2 = matrix(rnorm(nsub*ngene),nsub,ngene)
data2[1:4,] = data2[1:4,] + 2.2
```

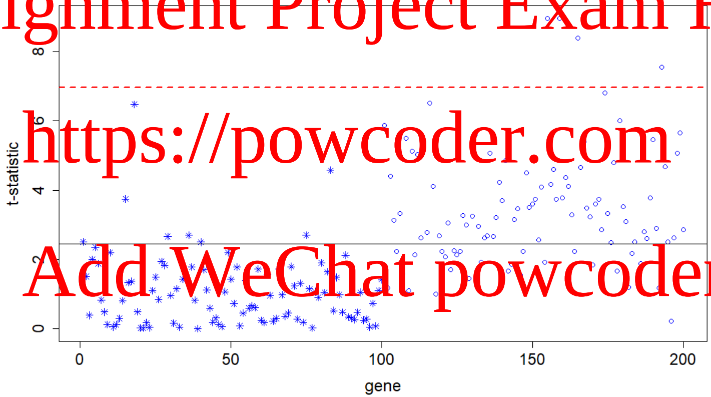
But only 4% of these genes are listed as significant after controlling for FWER (about 70% true positives with usual threshold).

Less bad than Bonferroni correction, but gets worse with number of obs.

Formally, we repeat the permutation procedure.

Power after FWER

Using standard threshold: 72 real discoveries, 8 false discoveries



Controling FWER: 4 real discoveries, 0 false.

But here half are real! Usually much less.

Modern Data

Micro-array experiments (1990's)

- Measure RNA Expression levels on 15,000 - 25,000

Modern OMICS data

- single nucleotide polymorphisms (SNP): 1,500,000 positions.

Want to relate to phenotype (cancer development/nose size).

Typically, a few tens or hundreds of genes/SNPs may be relevant; very hard to find needles in the hay.

But maybe FWER is too harsh?

False Discovery Rates

Suppose we are prepared to accept a few wrong conclusions in return for more power.

> *Here is a list of genes associated with eye color, we expect that 10% of this list are there by accident.*

Generally, we have more genes (I'll use k in notation below)

```
ngene = 1000
data = matrix(rnorm(nsub*ngene),nsub,ngene)
```

and examine p-values (could also do this with permutations)

```
gene.p.stat = function(matrix){
 ngene = ncol(matrix)
 p.gene = rep(0,ngene)
 for (i in 1:ngene){
   p.gene[i] = t.test(matrix[1:4,i],matrix[5:8,i])$p.value
  }
  return(p.gene)
}
```

With Some Non-Null Data

Typically maybe 5% of columns have real differences

```
data[5:8,1:50] = data[5:8,1:50] + 4
p.obs = gene.p.stat(data)
```

With no corrections we get

```
> sum(p.obs<0.05)
[1] 93
> sum(p.obs[1:50]<0.05)
[1] 50
```

So proportion of "discoveries" that are false is

```
> sum(p.obs[51:1000]<0.05)/sum(p.obs<0.05)
[1] 0.4623656
```

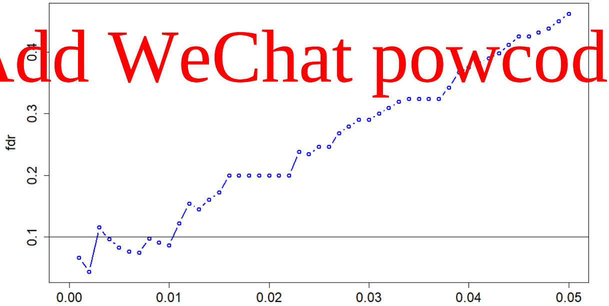
After Bonferroni, only 2 (real) discoveries!

```
> sum(p.obs<0.05/ngene)
[1] 2
```

What About Other Thresholds?

Look over thresholds from 0 to 0.05:

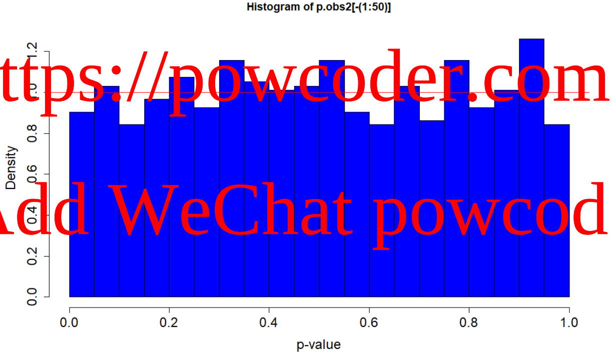
```
cuts = seq(0,0.05,by=0.001)
fdp = rep(NA,length(cuts))

for(i in 1:length(cuts)){
 fdp[i] = sum(p.obs[51:1000]<cuts[i])/sum(p.obs<cuts[i])
}
```



When You Don't Know the Truth

False discovery proportion required us to know which genes are non-null!

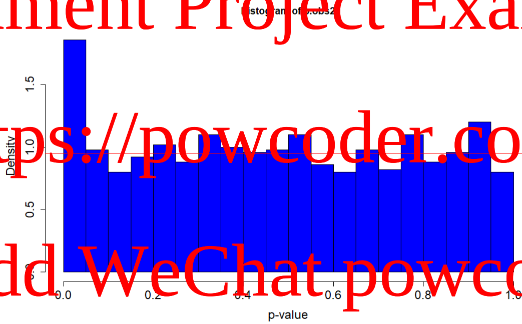
But, we do know that the null genes have uniform p-values:



So for a threshold q we expect to see kq null genes with p-values less than q.

When You Don't Know the Truth

But some real genes add an excess of small p-values (5% in our simulated data).



So for any q with

- m p-values less than q
- expect kq are null $m - kq$ non-null
- False Discovery Rate is kq/m = *expected* proportion of false discoveries.

Calculating False Discovery Rates

We want a list of genes, so we'll consider cut-offs at each p-value.

First sort the p-values smallest to largest

```
p.sort = sort(p.obs)
```

The position tells us # of genes up to each p-value

```
m = 1:ngene
```

So FDR for *each gene* is

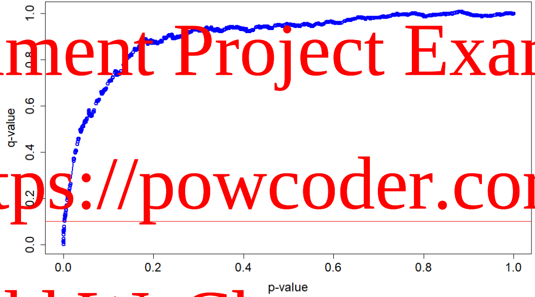
```
q.vals = ngene*p.sort/m
```

And we choose cutoff q-value at 0.1

```
cut = max( p.sort[q.vals<=0.1])
```

In this case we need a p-value less than about 0.002.

Results in Simulated Data



We find

```
> sum( p.obs[1:50] < cut )
[1] 21
> sum( p.obs < cut )
[1] 22
```

Or an FDP of $1/22 = 0.045$ in this case.

Variation

Many variations on FDR:

- Estimate number of nulls first.
- Look at a small range of p-values (not just (0, cut)).

We could look at how many p-values bigger than 0.5: about twice that number should be null

```
> k = 2*sum(p.obs > 0.5)
[1] 950
```

Then I can use this rather than ngene

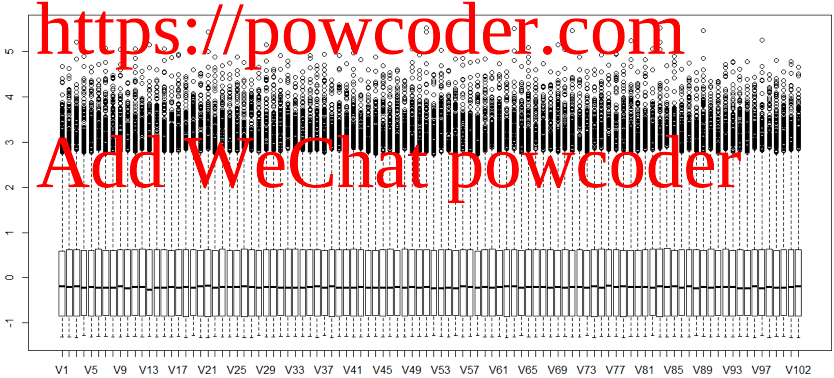
```
q.vals2 = k*p.sort/m
```

Increases to 25 genes with 22 real discoveries, FDP = 3/25 = 0.12.

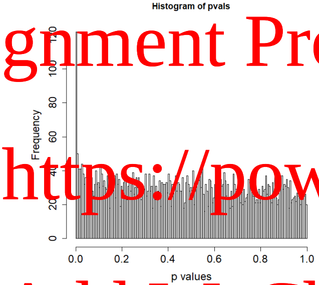
Some Real Data

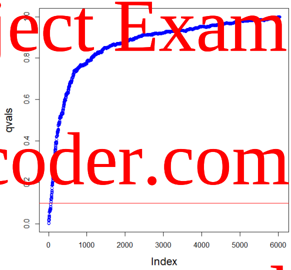
Expression levels of 6033 genes in 50 men with prostate cancer and 52 men without.

Expression Levels Over Users



t-Statistic P-values and Associated Q-values





And we can generate a list of genes of interest.

```
> cut = max(p.sort[qvals<=0.1])
> which(pvals<cut)
 [1]    2   11  332  364  579  610  694  698  702  721  735  739  905  914  921
[16] 1068 1077 1089 1113 1130 1314 1346 1557 1588 1589 1720 2370 2856 2897 2945
[31] 3017 3260 3282 3292 3375 3505 3600 3647 3665 3930 3940 3991 4000 4040 4073
[46] 4088 4104 4154 4316 4331 4396 4518 4546 4549 4552 4981
```

Summary

Multiple testing is a primary (not sole) cause of replication crisis in science.

Arises in multiple contexts:

- Direct (as here) testing of many effects.
- Selecting covariates in a linear model.
- Choosing hypotheses that appear strong
- Collecting data until significance is reached.
- Choosing outcomes to measure.

Possible remedies:

- Find a maximal statistic
- Bonferroni or other corrections (some equivalent to max statistics)
- False discovery rates

What is appropriate depends on your purpose.