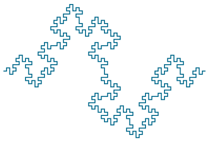
Duke Microbiome Immunology Cancer (MIC) Course

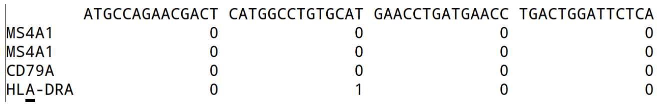
Elements of Statistical Inference

Biostatistics and Bioinformatics



Summer 2022

Single-cell RNA-Seq: Count matrix

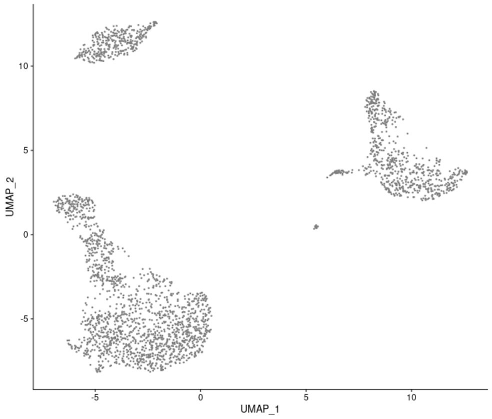


```
          ATGCCAGAACGACT CATGGCCTGTGCAT GAACCTGATGAACC TGACTGGATTCTCA
MS4A1                  0              0              0              0
MS4A1                  0              0              0              0
CD79A                  0              0              0              0
HLA-DRA                0              1              0              0
```

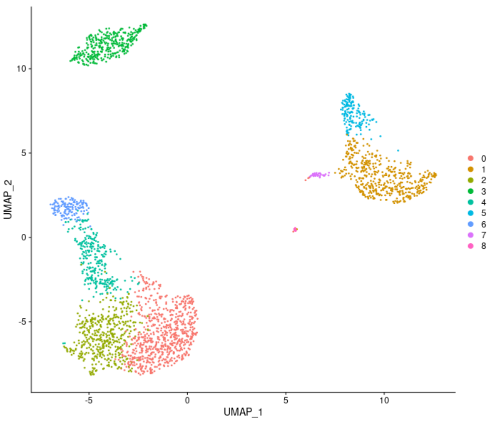
$$
\begin{array}{c|lllll}
 & & & \textbf{Cells} & & \\
 & & \text{cell } 1 & \text{cell } 2 & \dots & \text{cell } n \\
 & \text{gene } 1 & K_{11} & K_{12} & \dots & K_{1n} \\
\textbf{Genes} & \text{gene } 2 & K_{21} & K_{22} & \dots & K_{2n} \\
 & \vdots & \vdots & \vdots & \dots & \vdots \\
 & \text{gene } m & K_{m1} & K_{m2} & \dots & K_{mn}
\end{array}
$$

Table: K_{ji} number of reads mapped to gene j in cell i

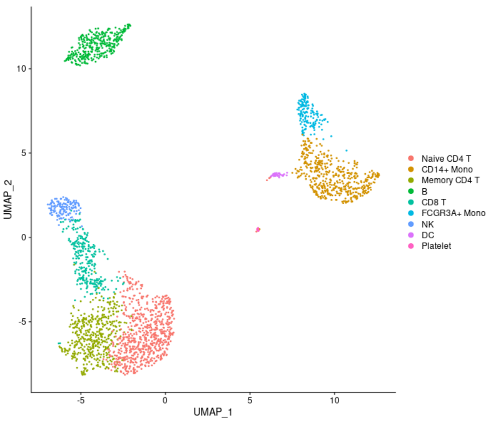
Single-cell RNA-Seq: UMAP



Single-cell RNA-Seq: UMAP (with nine inferred cluster)

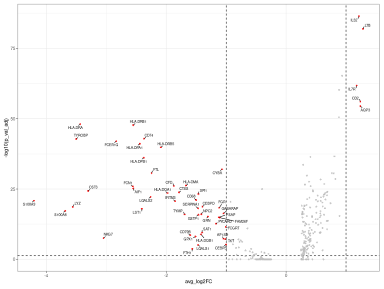


Single-cell RNA-Seq: UMAP (with inferred cluster types)



Single-cell RNA-Seq: Differential Expression Analysis

	p_val	avg_log2FC	pct.1	pct.2	p_val_adj
IL32	2.594e-91	1.215	0.949	0.466	3.557e-87
LTB	7.994e-87	1.283	0.981	0.644	1.096e-82
CD3D	3.922e-70	0.936	0.922	0.433	5.379e-66
IL7R	1.131e-66	1.178	0.748	0.327	1.551e-62
LDHB	4.082e-65	0.884	0.953	0.614	5.598e-61
CD2	5.526e-61	1.239	0.657	0.245	7.579e-57



Outline

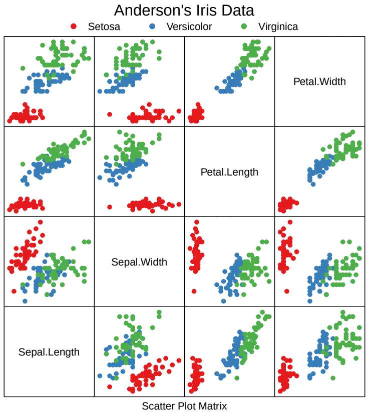
- Unsupervised learning
- Hypothesis testing
- Effect size estimation
- Multiple testing

Section 1

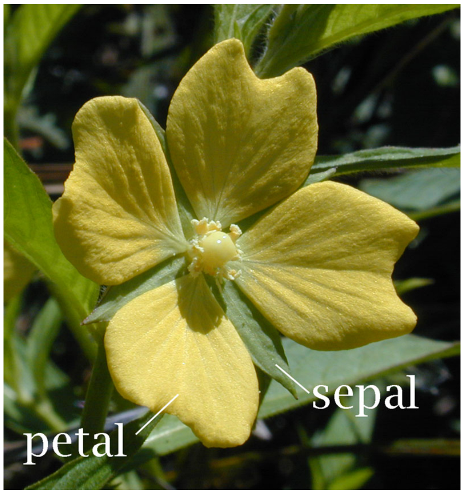
Unsupervised Learning

Anderson's Iris Data

$n = 150$ iris samples (Edgar A. The irises of the Gaspe peninsula. *Bulletin of the American Iris Society.* 1935; 59: 2–5)

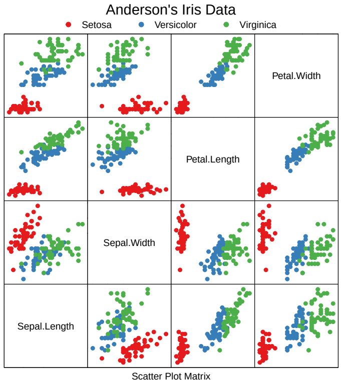


Petals and Sepals



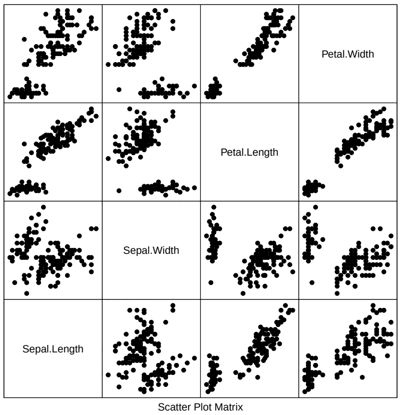
https://en.wikipedia.org/wiki/Sepal

Anderson's Iris Data

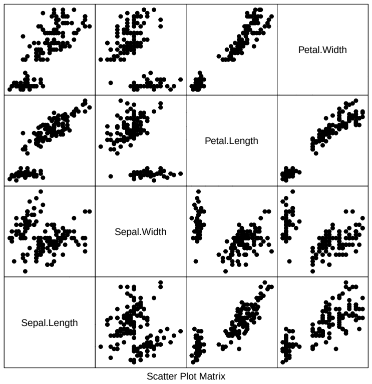


- Features (Observable; $m = 4$ variables): Petal width, petal length, sepal width, sepal length
- Class (observable; $k = 3$ levels): setosa, versicolor, virginica

Anderson's Iris Data (blinded)



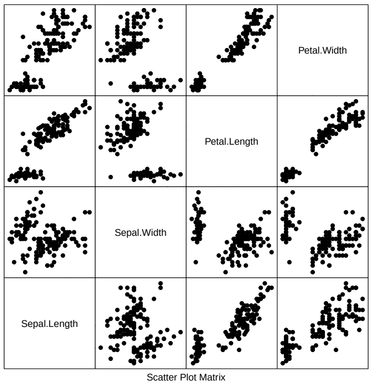
Anderson’s Iris Data (blinded)



- Features (Observable; $m = 4$ variables): Petal width, petal length, sepal width, sepal length
- Class (latent; $k = 3$ levels): setosa, versicolor, virginica

Anderson’s Iris Data (blinded)

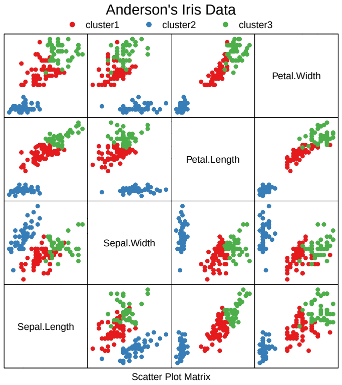
More realistic (the number of classes/clusters is not known)

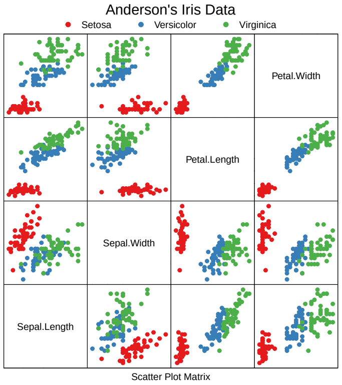


- Features (Observable; $m = 4$ variables): Petal width, petal length, sepal width, sepal length
- Class (latent; $k =?$ levels): setosa, versicolor, virginica
- k could be as small as 1
- technically speaking as large as $n = 150$ (each flower is its own class)

Anderson’s Iris Data (blinded)

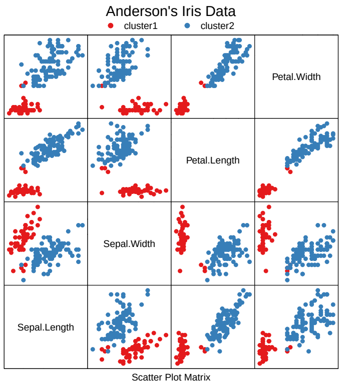
Assume that $k = 3$ is known (unrealistic in most applications)

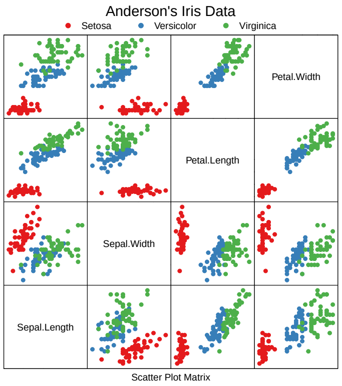




ANDERSON'S IRIS DATA (BLINDED)

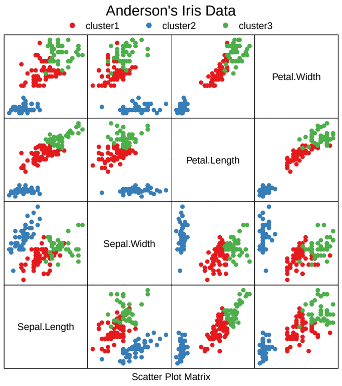
k is unknown; ask for $k = 2$ clusters

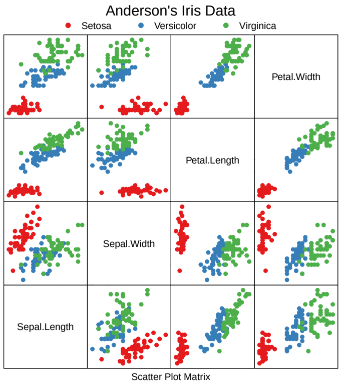




ANDERSON'S IRIS DATA (BLINDED)

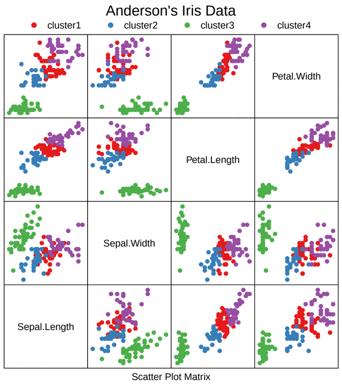
k is unknown; ask for $k = 3$ clusters

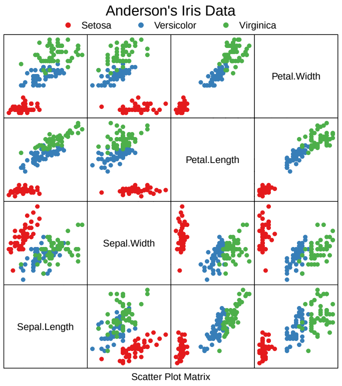




ANDERSON'S IRIS DATA (BLINDED)

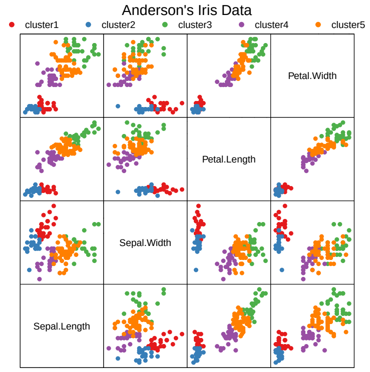
k is unknown; ask for $k = 4$ clusters



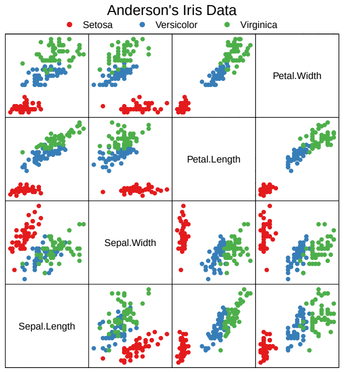


Anderson's Iris Data (blinded)

k is unknown; ask for $k = 5$ clusters



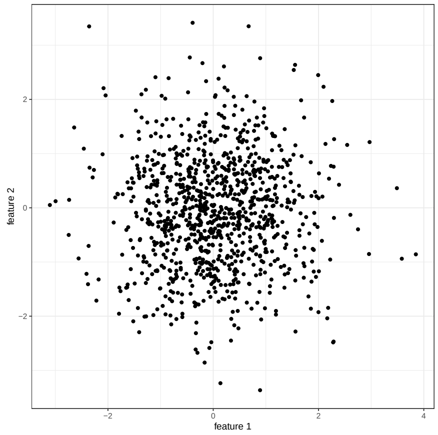
Scatter Plot Matrix



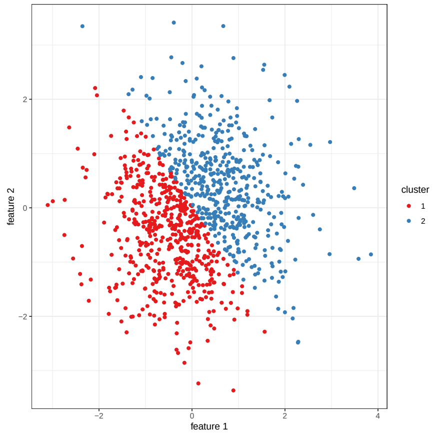
Scatter Plot Matrix

Another Example: Infer clusters

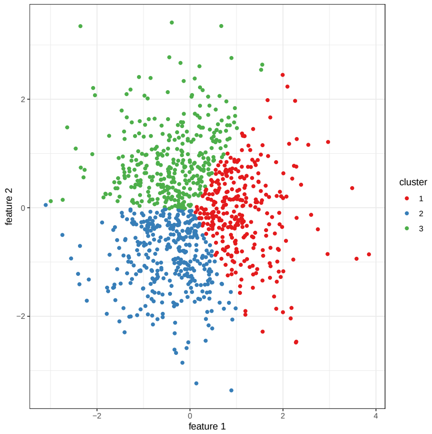
This is a simulated example consisting of two features. I am keeping the number of clusters secret.



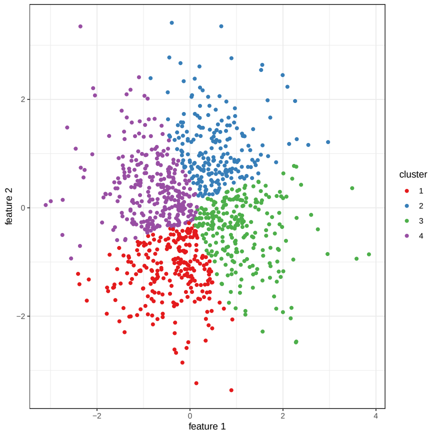
Another Example: $k = 2$



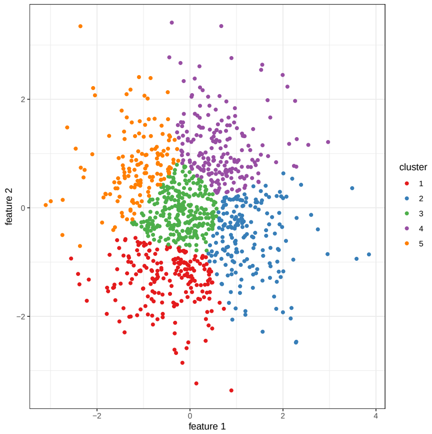
Another Example: $k = 3$



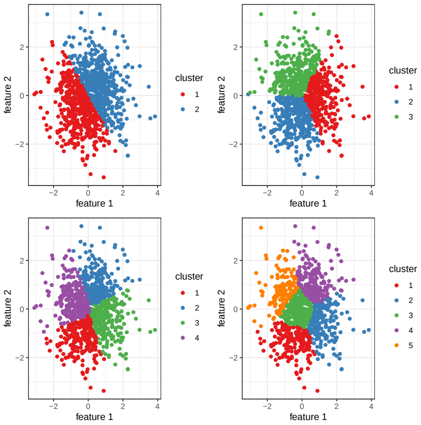
Another Example: $k = 4$



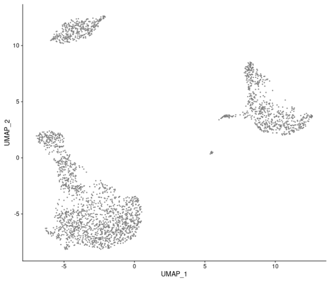
Another Example: $k = 5$



Another Example: Compare $k = 2, 3, 4$ and 5



scRNA UMAP



- n cells and m genes that passed QC filters
- Observable: a vector of counts of m genes for each of the n cells
- Is the gene expression profile for each cell observable?
- The number of cell types k is unknown
- The cell type for each cell is a latent variable (to be inferred)
- Number of cell types in the tumor may not be equal to k.

Unsupervised Learning

- Let X denote the genetic/genomic profile of a sample (or cell)
- Often we would like to discover groups, clusters or outliers based on the genetic profiles of the samples (cells in case of scRNA-Seq)
- These are *unsupervised* methods in the sense that the algorithm knows nothing about the grouping/clustering
- The method is only aware of the genetic profile (X) and not the phenotype Y
- The goal is to infer latent/hidden phenotypes using the observable features

Supervised Learning

- The features X are the phenotype Y are both observable in a "training" data set
- A model to predict
- The building of this model is "supervised" by the observed phenotype in the training data set
- Once the model has been finalized, its performance is assessed by applying it to a test/validation data set
- Only the features are available in the latter.

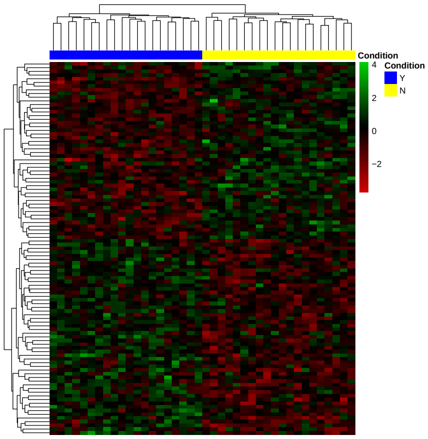
Unsupervised Analysis?

- Select a panel of genes based on the two-sample t-test
- Construct a panel of these "top hits"
- Carry out clustering with respect to the samples (the columns)
- Carry out clustering with respect to the genes in the panel (the rows)
- Present the results using a heatmap

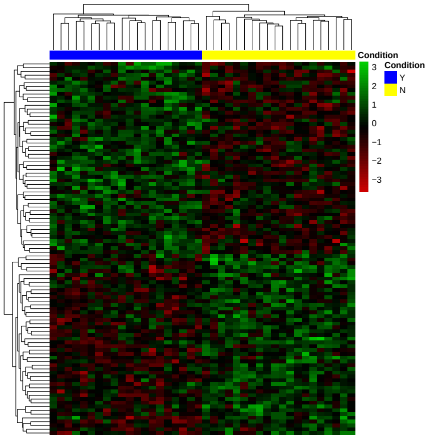
R Code to simulate Heatmap

```r
simulate.noise.heatmap <- function(n, m, alpha) {
    #' Simulate Expression Matrix: m gene and n+n=2n cases
    EXPRS <- matrix(rnorm(2 * n * m), m, 2 * n)
    #' Arbitrarily assign the first n cases to group 0
    #' and the remaining n cases to group 1
    grp <- factor(rep(0:1, c(n, n)))
    #' Assign dummy gene and case ids
    rownames(EXPRS) <- paste("Gene", 1:m, sep = "")
    colnames(EXPRS) <- paste("case", 1:(2 * n), sep = "")
    #' Get the two sample t-statistics for each of the m genes
    pvals <- genefilter::rowttests(EXPRS, grp)$p.value
    #' Pick genes whose corresponding P-value < alpha
    topgenes <- which(pvals < alpha)
    EXPRS <- EXPRS[topgenes, ]
    #' Produce an annotated heatmap
    annodat <- data.frame(Condition = ifelse(grp == 0, "N", "Y"), row.names = colnames(EXPRS))
    pheatmap::pheatmap(EXPRS, border_color = NA, show_rownames = FALSE, show_colnames = FALSE,
        annotation_col = annodat, color = colorRampPalette(c("red3", "black",
            "green3"))(50), annotation_colors = list(Condition = c(Y = "blue",
            N = "yellow")))
}
```

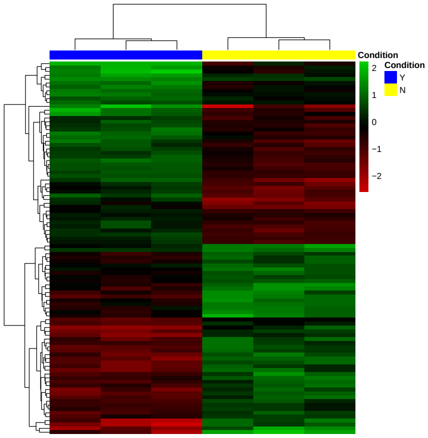
Heatmap Example: $m = 20,000, n = 20,\ \alpha = 0.005$



Heatmap Example: $m = 40,000, n = 20,\ \alpha = 0.0025$



Heatmap Example: $m = 20,000, n = 3,\ \alpha = 0.005$



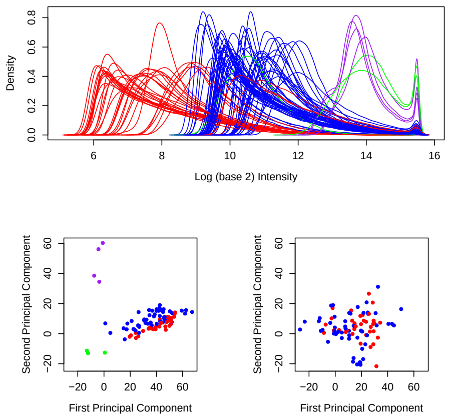
Semi-supervised Learning

- Some consider this an *unsupervised* analysis as the clustering algorithm is unaware of the classes
- This is not an accurate assessment: It is actually a supervised analysis in the sense that we are picking the top hits based on the phenotype
- A procedure is *unsupervised* if the class info is only used for annotation of the final figure
- Keep this in mind when reviewing papers presenting claims based on observations from heatmaps
- Side note: A similar caveat is present in pathway analyses, where investigators limit the analysis to genes in the top hit panel.

Batch Effect Discovery

- Clustering methods are very useful for detecting batch effects in genomic data
- Batch effects tend to be stronger that biological effects
- These often affect most genes (the biological effect may only be captured by a few)
- This can be an effective weapon in your QC arsenal (this is how I start any new analysis)

From CCR 2008 Paper

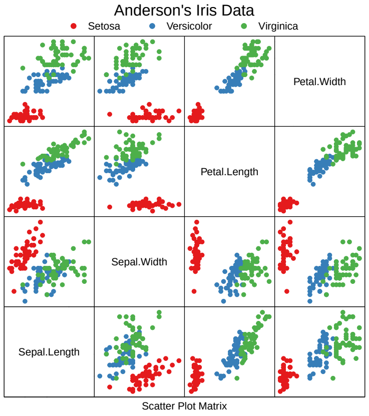


Dimension reduction, feature selection/extraction

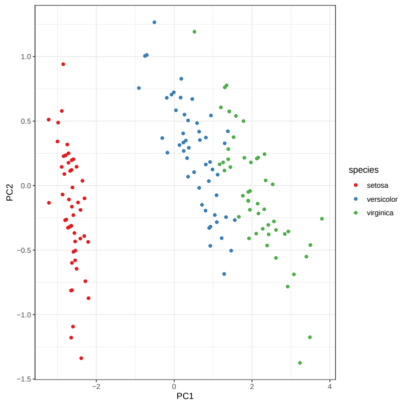
- The Anderson data consisted of only four features
- The number of features in a whole transcriptome analysis (the dimension) is substantially larger (50-60K)
- Dimension reduction is a key step in these analyses
- Criteria for reduction
  - Parsimony/reduce redundancy (pathological example: no need to measure temperature in Celsius, Fahrenheit and Kelvin)
  - Select features that explain greatest variability (pathological example: a feature constant across all cases is not useful)

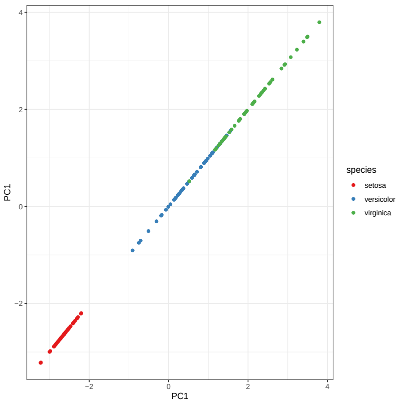
Revisit Anderson's Data

Note that there is substantial correlation among the 4 features (redundancy)



Revisit Anderson's Data





A Self-fulfilling Prophecy

- Statistical methods for unsupervised learning guarantee one thing
- They will return a clustering of your data
- What they do not guarantee and are invariably unable to verify, is the biological relevance or reproducibility of the clustering
- In light of this Self-fulfilling Prophecy, these methods should be used with utmost care
- Methods for “optimal” clustering are under active development (including by faculty in our dept)

Section 2

Hypothesis Testing

Hypothesis Testing: A Generic Overview

- Formulate a scientific hypothesis (conceptual)
- Formulate a corresponding statistical hypothesis (quantitative)
- Specify an experimental design
- Specify the decision procedure:
    - an appropriate test statistic
    - decision rule based on the test statistic (typically under a set of assumptions)
- Execute Experiment (collect data)
- Apply the decision procedure to the realized outcomes of the experiment
- Draw a conclusion as to the level of empirical evidence in support of the posited statistical hypothesis

Hypothesis Testing: Null versus Alternative

- There are two hypotheses: null (H_0) and alternative (H_1)
- The null hypothesis posits the status quo
- H_0 is the conservative hypothesis
- In the US legal system, the defendant is presumed to be innocent
- The null hypothesis: Defendant is innocent
- Study: Investigate if gene XYZ is differentially expressed with respect to treatment
- Corresponding hypotheses:
  - H_0 : gene XYZ is *not* differentially expressed with respect to treatment
  - H_1 : gene XYZ is differentially expressed with respect to treatment

More on Null versus Alternative

- Suppose that your are studying the effect of a drug in a clinical study
- Safety Study:
  - H_0: Drug is toxic
  - H_1: Drug is safe
- Efficacy study:
  - H_0: Drug is not efficacious
  - H_1: Drug is efficacious

Hypotheses of common interest in scRNA studies

- Is a gene differentially expressed with respect to a given cluster (against the cells in all the other clusters)
- Within a given cluster, is a gene differentially expressed with respect to a treatment or a phenotype
- Does the differential expression effect with respect to a treatment or a phenotype depend on the cluster (interaction hypothesis)

Decision

- Based on the empirical evidence using the decision rule, we will
  - either reject the null hypothesis H_0 in favor of H_1
  - or fail to reject H_0 (an inconclusive outcome)
- IMPORTANT: Failing to reject H_0 does *not* afford us to conclude that H_0 is *true*
- There is a longstanding controversy with respect to making decision based on $P < 0.05$
- Making decision based on $P > 0.05$ is more egregious
- The earth is flat ($p > 0.05$): significance thresholds and the crisis of unreplicable research (Amrhein *et al.*,; PeerJ. 2017)

Notation: True versus False Null Hypothesis

- The truth may be stated either by the null or alternative hypothesis
- If the truth is stated by the statement of the null hypothesis, we will say that
  - The null hypothesis is true
  - or call it a true null hypothesis
- If the truth is stated by the statement of the alternative hypothesis, we will say that
  - The null hypothesis is false
  - or call it a false null hypothesis
- We will use these terms for notational convenience

Dichotomizing the truth and decision

- Dichotomy on truth: H_0 or H_1
- Dichotomy on decision: Reject H_0 or Fail to reject H_0
- The decision will result in one of four outcomes
- Two of these will be correct and the other two will be erroneous decisions

Type I and II errors

- Type I Error: Erroneously decide in favor of the alternative hypothesis (reject a true null hypothesis)
- Type II Error: Erroneously decide in favor of the null hypothesis (fail to reject a false null hypothesis)
- The so called ”alpha” level is the probability of a type I error
- The ”power” of a test, is the complement of the probability of the type II error
- IMPORTANT: There is a trade-off between these two error rates

Type I and II errors

		Null Hypothesis (H_0)	
		True	**False**
Decision	**Fail to reject H_0**	Correct Decision	Type II error
	Reject H_0	Type I error	Correct Decision

Type I and II error trade-off

- In our court system, a defendant is presumed innocent until proven guilty
  - Type I error: Convict an innocent defendant
  - Type II error: Fail to convict a guilty defendant
- If the prosecution gets too much leeway, the the likelihood of convicting an innocent defendant increases
- Conversely, if the prosecution is reigned in by the judge, the likelihood of letting a guilty defendant walk free increases
- Similar analogy in the case of a smoke detector:
  - Dialing up the sensitivity, increases the likelihood of annoying beeps (false alarms) when using your toaster
  - Dialing down the sensitivity, increases the likelihood of missing a true fire

Notation: Decision

- false-positive (FP): Reject a true null hypothesis (Type I error)
- true-positive (TP): Reject a false null hypothesis
- false-negative (FN): Fail to reject a false null hypothesis (Type II error)
- true-negative (TN): Fail to reject a false null hypothesis

		Null Hypothesis (H_0)	
		True	**False**
Decision	**Fail to reject H_0**	TN	FN
	Reject H_0	FP	TP

Three Decision Rules

- Following the collection of data, consider using one of the three decision rules
- Decision Rule 1: Reject H_0
- Decision Rule 2: Do not reject H_0
- Decision Rule 3: Flip a coin: Reject H_0 if tails and do not reject H_0 if heads
- Note that each of the three decision rules ignores the data.
- What are the type I and II error rates for these decision rules?
- Which one would you choose?

Decision Rule 1 (Always Reject H_0)

- If H_0 is true, then it will be rejected
- A false-positive decision will be made if H_0 is true
- $\alpha = 1$
- If H_0 is false, then it will be rejected
- A true-positive decision will be made if H_0 is false
- $\beta = 0$

Decision Rule 2 (Do not Reject H_0)

- If H_0 is true, then it will not be rejected
- A false-positive decision will not be made
- $\alpha = 0$
- If H_0 is false, then it will not be rejected
- A false-negative decision is will be made
- $\beta = 1$

Decision Rule 3 (Flip a coin)

- If H_0 is true, then the probability of rejecting it is one-half
- $\alpha = \frac{1}{2}$
- If H_0 is false, then probability of not rejecting it is one-half
- $\beta = \frac{1}{2}$

A Bad Rule is a Valid (but bad) Decision Rule

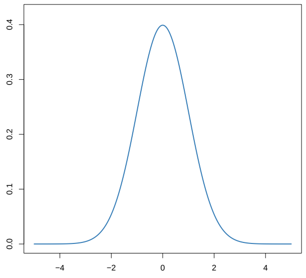
Decision	**Description**	α	β
1	Always reject H_0	1	0
2	Always accept H_0	0	1
3	Flip a coin	$\frac{1}{2}$	$\frac{1}{2}$

- Note that these decision rules effectively ignore the data
- While they are poor decision rules, they are technically valid decision rules
- A poor statistical approach will effectively reduce to one of these three
- Note that while $\alpha + \beta = 1$ in all these cases, that is generally not the case
- The type I error is generally *not* the complement of the type II error

Quick Note: Conservative versus Anti-conservative; Robustness

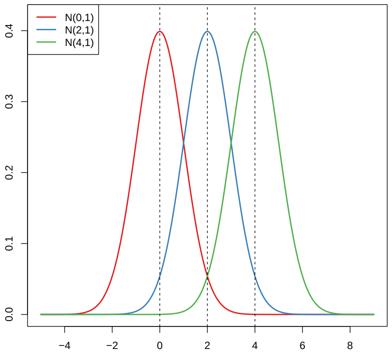
- The properties of the decision rule will depend on underlying assumptions
- They may be greatly sensitive to these assumptions
- The type I error of a decision procedure we hope to achieve is called the *nominal* level
- Example: If we claim that the nominal level of our decision is 0.05, then we are *claiming* that the probability of committing a false-positive is at most 0.05.
- If the *actual* type I error rate exceeds the nominal level the test is said to be anti-conservative
- If the *actual* type I error rate is less than the nominal level the test is said to be conservative
- A decision rule that is not sensitive to the underlying assumptions, with respect to type I error control, is said to be robust

Normal Distribution

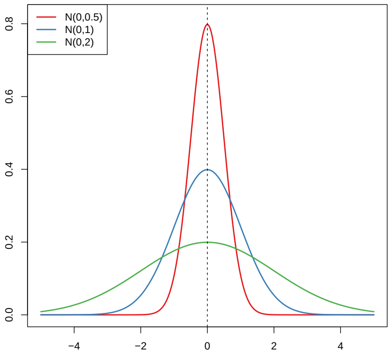


- N(μ,σ) denotes a normal distribution with mean μ and standard deviation σ
- The mean parameter determines the center of the distribution
- The standard deviation controls the spread around the mean

Normal Distribution: Shifting the mean



Normal Distribution: Shifting the variance



The Normal two-sample problem

- The expression of gene XYZ in group 1 (*e.g.*, wild-type) follows N(μ_1,σ)
- The expression of gene XYZ in group 1 (*e.g.*, mutant) follows N(μ_2,σ)
- Under H_0 the distribution does not depend on the group
- As we have *assumed* that the distributions are normal *and* the standard deviation are equal, H_0 is equivalent to $\mu_1 = \mu_2$
- Differential expression hypothesis: $H_0 : \mu_1 = \mu_2$ versus $H_1 : \mu_1 \neq \mu_2$
- The alternative is a shift in the mean
- We will test this hypothesis using the two-sample t-statistic

Simulation Example: Function

```
simttest <- function(n1, n2, mean1, mean2, stdev1, stdev2, alpha) {
    ## Simulate n1 observations from group 1 N(mu1,stdev1)
    y1 <- rnorm(n1, mean1, stdev1)
    ## Simulate n2 observations from group 2 N(mu2,stdev2)
    y2 <- rnorm(n2, mean2, stdev2)
    ## Perform two-sample unpaired t-test assuming equal variance
    testresult <- t.test(y1, y2, paired = FALSE, var.equal = TRUE)
    ## Get P-value of test
    pvalue <- testresult$p.value
    ## Apply decision rule: Reject if pvalue < alpha
    reject <- ifelse(pvalue < alpha, TRUE, FALSE)
    ## Return decision
    return(reject)
}
```

Simulation Example: Simulate under H_0

- A two-sample experiment with $n = 3$ in each group
- B =10 simulation replicates under H_0:
    - $n_1 = 3, n_2 = 3$
    - $\mu_1 = 1 = \mu_2 = 1$
    - $\sigma_1 = \sigma_2 = 0.5$

```
set.seed(124228)
res <- replicate(B, simttest(3, 3, 1, 1, 0.5, 0.5, alpha = 0.05))
tibble::tibble(experiment = 1:B, decision = res) %>%
    kableExtra::kbl() %>%
    kableExtra::kable_classic()
```

experiment	decision
1	FALSE
2	FALSE
3	FALSE
4	FALSE
5	FALSE
6	FALSE
7	FALSE
8	TRUE
9	FALSE
10	FALSE

There is 1 type I error (false-positive)

Simulation Example: Simulate under H_1

- A two-sample experiment with $n = 3$ in each group
- B =10 simulation replicates under H_1:
    - $n_1 = 3, n_2 = 3$
    - $\mu_1 = 1 \neq \mu_2 = 2$
    - $\sigma_1 = \sigma_2 = 0.5$

```
set.seed(515721)
res <- replicate(B, simttest(3, 3, 1, 2, 0.5, 0.5, alpha = 0.05))
tibble::tibble(experiment = 1:B, decision = res) %>%
    kableExtra::kbl() %>%
    kableExtra::kable_classic()
```

experiment	decision
1	TRUE
2	TRUE
3	TRUE
4	TRUE
5	FALSE
6	TRUE
7	TRUE
8	FALSE
9	TRUE
10	TRUE

There are 2 type II errors (false-negatives)

Simulation Example: Type I error

The type I error probability can be estimated by the empirical rejection rate over a large number of replicate experiments

```
set.seed(536234)
B <- 10000L
mean(replicate(B, simttest(3, 3, 1, 1, 0.5, 0.5, 0.05)))

## [1] 0.0497
```

Simulation Example: Type I error

Why is the empirical type I error rate above the nominal level in the following examples

```
set.seed(51621)
B <- 10000L
mean(replicate(B, simttest(3, 3, 1, 1, 0.5, 1, 0.05)))

## [1] 0.065
```

```
mean(replicate(B, simttest(3, 3, 1, 1, 0.5, 2, 0.05)))

## [1] 0.0841
```

Designing a two-sample experiment

- The sample size to achieve the desired power at a given type I error rate depends on the effect size
- Given everything else fixed, a larger effect size requires a smaller size to achieve a power at a given type I error rate
- The effect size for the two-sample t-test is defined as

$$\Delta = \frac{|\mu_0 - \mu_1|}{\sigma}$$

- The numerator $|\mu_0 - \mu_1|$ is the difference (in absolute value) of the means
- The size of this difference (how large it is) is in relation to (scaled by) the standard deviation
- Under H_0: $\Delta = 0$.

Example

Suppose that

- $\mu_1 = 0$
- $\mu_1 = 2$
- $\sigma = 1$

The standardized effect size is

$$\Delta = \frac{|\mu_0 - \mu_1|}{\sigma} = \frac{|2 - 0|}{1} = 1$$

Suppose that you want to have a power of 90% to detect this effect size at the $\alpha = 0.05$ level using the two-sample t-test.

Forget about the Design

What is the power if we use 3 units per group

```
des <- power.t.test(n = n, delta = abs(2 - 1), sd = 1, sig.level = 0.05)
des

##
##      Two-sample t test power calculation
##
##               n = 3
##           delta = 1
##              sd = 1
##       sig.level = 0.05
##           power = 0.1572361
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

The type II error rate is 0.84!

Forget about the Design

What is the power if we use 6 units per group

```
des <- power.t.test(n = n, delta = abs(2 - 1), sd = 1, sig.level = 0.05)
des

##
##      Two-sample t test power calculation
##
##               n = 6
##           delta = 1
##              sd = 1
##       sig.level = 0.05
##           power = 0.3471565
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

While improved (by virtue of increasing sample size), the type II error rate is 0.65.

Now Use Experimental Design

- The required sample size, per group, to detect an effect size of
$$\Delta = \frac{|0-2|}{1} = 1$$
with a power of 0.9, at the 0.05 level is $n =$23 *per* group.

```
##
##      Two-sample t test power calculation
##
##               n = 22.0211
##           delta = 1
##              sd = 1
##       sig.level = 0.05
##           power = 0.9
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

- If a smaller sample size is used, the study will be under-powered
- What is the caveat with using a larger sample size?
- Note: These observations are based on the given assumptions, effect size and type I and II errors

Simulation Example: Verify Power

To verify the power empirically using the principles you have learned (without using a sample size formula).

```
set.seed(91921)
B <- 10000L
mean(replicate(B, simttest(22, 22, 2, 1, 1, 1, 0.05)))

## [1] 0.8981

mean(replicate(B, simttest(23, 23, 2, 1, 1, 1, 0.05)))

## [1] 0.9166
```

To get the power, estimate the rejection rate under H_1

Simulation: Important Notes

- Data are generated under the truth
- Parameters and distributions are set by you
- A simulated experiment is to mimic a hypothetical, but real, experiment
- The truth is not known in the context of a real experiment
- IMPORTANT: The decision rule step has to remain *blinded* to this truth
- Computing Exercise: Evaluate the type I error and power for the two-sample example using simulation and formula

Experimental Design

- Two examples
  - Decide upfront to evaluate $n = 10$ experimental units
  - Decide to initially evaluate $n_1 = 5$ experimental units (Stage 1). Depending on the results evaluate an additional $n_2 = 5$ experimental units
- These are *different* experimental strategies
- Design 1: The final sample size is $n = 10$
- Design 2: The final sample size is $n = n_1 = 5$ *or* $n = n_1 + n_2 = 10$
- The statistical properties of your decision rule depends on the strategy used.

Statistical versus Clinical/Biological Significance

- Hypothesis testing is carried out to investigate *statistical* and not *biological* significance
- It is the responsibility of the investigator to pose a biologically relevant hypothesis.
- It is also the responsibility of the investigator to ensure that a statistically significant finding is biologically plausible/realistic
- Statistical significance does not necessarily imply biological significance or vice versa

Biologically but not Statistically Significant

```
set.seed(1122333)
x0 <- rnorm(3, 1, 1)
x1 <- rnorm(3, 2, 1)
x0

## [1] -0.25824011  0.02820527  2.20878939

x1

## [1] 1.5462733 0.6578732 3.1782064

t.test(x0, x1)

##
##  Welch Two Sample t-test
##
## data:  x0 and x1
## t = -1.0572, df = 3.9884, p-value = 0.3502
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.117361  1.848295
## sample estimates:
## mean of x mean of y
## 0.6595849 1.7941176
```

Statistically but not Biologically Significant

```
x0 <- c(3.0001, 3.0002, 3.0003, 3.0004, 3.0005)
x1 <- c(3.0006, 3.0007, 3.0008, 3.0009, 3.001)
x0

## [1] 3.0001 3.0002 3.0003 3.0004 3.0005

x1

## [1] 3.0006 3.0007 3.0008 3.0009 3.0010

t.test(x0, x1)

##
##  Welch Two Sample t-test
##
## data:  x0 and x1
## t = -5, df = 8, p-value = 0.001053
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.0007306004 -0.0002693996
## sample estimates:
## mean of x mean of y
##    3.0003    3.0008
```

Section 3

Estimation

Estimation

- The P-values quantifies the evidence in support of the statistical hypothesis
- It does not quantify the effect size
- What is often of interested is estimate the unknown parameters or quantities
- Examples
  - Mean level for the untreated group μ_0
  - Mean level for the treated group μ_1
  - Fold-change $\rho = \frac{\mu_1}{\mu_0}$
  - Standardized difference $\Delta = |\mu_1 - \mu_0|/\sigma$
- Two types of estimates
  - Point estimate
  - Interval estimate

Point Estimator

- A point estimator of μ is the so called sample mean
- The sample mean $\bar{x}_n$ is obtained by simply averaging all the observations
- Note that an alternative is to used the sample median (rather than sample mean)
- The sample median is obtained by first sorting the observations (in say ascending order)
- The median is the middle observation (among the sorted observation)
- The median is more robust against outliers

Confidence Intervals

- Example: The sample mean (the average of the observations) is a point estimate of the population (true) mean
- It is either equal to the true value of the parameter or is not
- As it is a single number it does not provide any direct measure of accuracy
- An interval estimate incorporates some measure of accuracy
- Thus it is generally more appropriate to present an interval estimate
- A common example of an interval estimate is the confidence interval

Covered or not covered

- The goal is to estimate μ
- If μ (the true but unknown parameter) is contained in the confidence interval, we say that it is "covered"
- Otherwise, it is not "covered"
- Note that when doing a simulation study, we can ascertain if μ is covered or not.
- Why?
- In real data analysis, we cannot ascertain if μ is covered by the confidence interval
- Why?
- We can only state that we are 95% *confident* that μ is covered by the interval estimate based on the data from our experiment
- More on "confidence" later

Simulate coverage

Suppose that the gene expression is distributed according to N(0,1). The following provides point estimator and 95% CI for μ based on 10 simulation replicates.

exp	n	mu	sigma	xbar	s	lcl	ucl	cover
1	6	0	1	0.3582991	0.2947594	0.0489681	0.6676302	FALSE
2	6	0	1	0.6721558	0.8578519	-0.2281046	1.5724161	TRUE
3	6	0	1	-0.2344397	0.6226090	-0.8878277	0.4189484	TRUE
4	6	0	1	-0.8755614	1.1545657	-2.0872039	0.3360810	TRUE
5	6	0	1	-0.8816406	0.7079849	-1.6246252	-0.1386560	FALSE
6	6	0	1	0.5730998	1.1525229	-0.6363989	1.7825984	TRUE
7	6	0	1	-0.0302625	1.5003950	-1.6048305	1.5443055	TRUE
8	6	0	1	-0.6153569	0.5388237	-1.1808176	-0.0498961	FALSE
9	6	0	1	-0.0464333	1.3454345	-1.4583802	1.3655136	TRUE
10	6	0	1	0.2099230	1.0739237	-0.9170908	1.3369369	TRUE

Confidence Interval: Common Misunderstanding

- A (not the) 95% CI for the mean based on the first experiment was $(0.05, 0.67)$
- A (not the) 95% CI for the mean based on the second experiment was $(-0.23, 1.57)$
- It is wrong to say that the probability that the first CI does not contain the true value $\mu = 0$ is 95%
- It is also wrong to say that the probability that the second CI contains the true value $\mu = 0$ is 95%
- We conduct one and only one experiment
- Based on the first experiment, we can say that we are 95% confident that it contains the true value
- Note that μ is *not* covered by the first experiment
- If we repeated the experiment a large number of times, 95% of the CIs would cover the true value
- We are 95% confident that the first experiment is among

Section 4

Multiple Testing

Introduction

- Analysis of high-dimensional data is concerned with assessing the statistical significance of multiple loci/genes
  - Microarray : 20,000-50,000 probe sets
  - GWAS: 500,000-5,000,000 typed SNPs
  - RNA-Seq: 25,000-60,000 genes/transcripts (humans)
- This leads to the *Multiple Testing* problem

Framework versus method

- It is important to distinguish between the framework (criterion) and method used to account for multiple testing
- One first has to decide which framework to use
- We will consider two widely-used frameworks: The family-wise error rate (FWER) and the false-discovery rate (FDR)
- Once the framework has been decided on, one has to pick an appropriate method to provide proper multiple testing control

Notation

- We plan to test m genes for differential expression
  - m_0 is number of genes not differentially expressed
  - m_1 is number of genes differentially expressed
  - $m = m_0 + m_1$
  - While m is known, m_0 and m_1 are unknown parameters
  - We assume that these are fixed parameters

Notation

- Corresponding to each of the m genes, there is a *marginal* null hypothesis
- H_j: Gene j is not differentially expressed
- We decide on a decision rule for each marginal hypothesis
- As in the single-gene case, when applied to gene j, the decision will be to either
  - reject H_j or
  - fail to reject (“accept”) H_j
- After applying the decision rule to all m genes
  - R will denote the number of marginal hypotheses rejected
  - A denotes the umber of marginal hypotheses accepted
  - R and A are observable random quantities
  - $m = A + R = m_0 + m_1$

Introduction: Summarizing a Multiple Testing Procedure

- The results from any multiple testing procedure can be summarized as the following table

	Accept	Reject	Total
Truth Null	A_0	R_0	m_0
Alt.	A_1	R_1	m_1
	A	R	m

- Notation:
  - m: Number of tests, m_0, m_1 number of null/true genes
  - R: Number of genes rejected according to the decision rule
  - A: Number of genes accepted according to the decision rule
  - R_0/R_1 number of TN/FP
  - A_0/A_1 number of FN/TP

Introduction: Example

- Results from an analysis based on $m = 10$ genes:

gene	truth	pvalue
gene1	0	0.29070
gene2	1	0.61630
gene3	1	0.00320
gene4	0	0.01641
gene5	0	0.25150
gene6	0	0.58450
gene7	0	0.22890
gene8	1	0.12630
gene9	0	0.26080
gene10	0	0.04980

- Investigator decides to use following decision rule: Any gene with a corresponding unadjusted P-value of less than 0.05 will be rejected.
- Reject H_j if $p_j < 0.05$ or accept H_j otherwise

Exercise: Fill in the 2x2 table

	Accept	Reject	Total
Truth Null	$A_0 =?$	$R_0 =?$	$m_0 =?$
Alt.	$A_1 =?$	$R_1 =?$	$m_1 =?$
	$A =?$	$R =?$	$m =?$

Example: Fill in the 2x2 table knowing the truth

@

- $m_0 = 7$ and $m_1 = 3$
- $R = 3$ will be rejected based on the decision rule
- Consequently $A = m - R = 7$ will be accepted
- $R_0 = 2, R_1 = 1, A_0 = 5$ and $A_1 = 2$
- Among the $R = 3$ rejections, there are $R_0 = 2$ false discoveries

Example: Fill in the 2x2 table knowing the truth

	Accept	Reject	Total
Truth Null	$A_0 = 5$	$R_0 = 2$	$m_0 = 7$
Alt.	$A_1 = 2$	$R_1 = 1$	$m_1 = 3$
	$A = 7$	$R = 3$	$m = 10$

gene	truth	pvalue
gene1	0	0.29070
gene2	1	0.61630
gene3	1	0.00320
gene4	0	0.01641
gene5	0	0.25150
gene6	0	0.58450
gene7	0	0.22890
gene8	1	0.12630
gene9	0	0.26080
gene10	0	0.04980

- $m_0 = 7$ and $m_1 = 3$
- $R = 3$ will be rejected based on the decision rule
- Consequently $A = m - R = 7$ will be accepted
- $R_0 = 2, R_1 = 1, A_0 = 5$ and $A_1 = 2$
- Among the $R = 3$ rejections, there are $R_0 = 2$ false discoveries

Example: Fill in the 2x2 table (real data analysis)

	Accept	Reject	Total
Truth Null	$A_0 =$	$R_0 =$	$m_0 =$
Alt.	$A_1 =$	$R_1 =$	$m_1 =$
	$A = 7$	$R = 3$	$m = 10$

gene	pvalue
gene1	0.29070
gene2	0.61630
gene3	0.00320
gene4	0.01641
gene5	0.25150
gene6	0.58450
gene7	0.22890
gene8	0.12630
gene9	0.26080
gene10	0.04980

Example: Fill in the 2x2 table (based on what we observe)

- We can only fill in the bottom row of the table

	Accept	Reject	Total
Truth Null	A_0	R_0	m_0
Alt.	A_1	R_1	m_1
	$A = 7$	$R = 3$	$m = 10$

- The remaining quantities are fixed unknown quantities or unobservable random variables.

Comments

	Accept	Reject	Total
Truth Null	A_0	R_0	m_0
Alt.	A_1	R_1	m_1
	A	R	m

- m is a known constant
- m_0 and m_1 are unknown constants
- R and A are determined on the basis of applying the decision rule to the data
- They are *observable* random quantities
- The true states of the genes of the genes are unknown
- A_0, A_1, R_0 and R_1 are *unobservable* random quantities

Introduction: Multiple Testing Problem

- Control error rate(s) in multiple testing context
- Multiple testing methods are designed to control a particular error rate
- Multiple error rates exist → need to chose error rate to control and then method to control it

Introduction: Error Rates

- **Family-wise error rate** (FWER): the probability of at least one type I error if all of the null hypotheses are true (*i.e.,* $m_0 = 0$)
- **False discovery rate** (FDR): the expected proportion of type I errors among the rejected hypotheses.

Family-wise Error Rate (FWER)

- Suppose that all m genes are null (*i.e.,* $m = m_0$ or $m_1 = 0$)
- In this case, ideally, you would not reject any of the m genes (*i.e.,* $R = 0$)
- If $R > 0$ (or equivalently $R \geq 1$), then at least one false-positive decision has been committed
- FWER is the probability of committing at least one false-rejection (among m) given that *all* genes are null

$$\text{FWER} = P(R \geq 1 | m = m0)$$

- Note that when $m = 1$ (single gene), this definition is identical to the type I error we have previously considered

Commonly used methods for FWER control

The following FWER control methods are provided by the `stats::p.adjust` function

- Bonferroni's Method
- Holm
- Hochberg
- Hommel
- Permutation resampling (provided by the Bioconductor `multtest` package)

Controlling FWER: Bonferroni Method

- In the single gene case, a standard decision rule is to compare the *P-value* against α
- The Bonferroni approach: Compare each of the m marginal *P-value*s to $\frac{\alpha}{m}$
- The Bonferroni adjusted *P-value* is defined as

$$P_j = m \times p_j$$

- Technical note: P_j, as defined above, could be larger
- If $m \times p_j$ is larger than 1, then truncate P_j at 1.

gene	pvalue	padj
gene1	0.29070	1.0000
gene2	0.61630	1.0000
gene3	0.00320	0.0320
gene4	0.01641	0.1641
gene5	0.25150	1.0000
gene6	0.58450	1.0000
gene7	0.22890	1.0000
gene8	0.12630	1.0000
gene9	0.26080	1.0000
gene10	0.04980	0.4980

False Discovery Rate (FDR)

- Consider the quantity $\frac{R_0}{R}$
- This is the proportion of of false discoveries among the genes rejected
- This is an *unobservable* random quantity (R_0 is not observable)
- In the FDR framework is based on controlling the *expected* value of this ratio
- FDR $\equiv E[\frac{R_0}{R}]$
- Note that when $m_0 = m$ (none of the genes are true), FWER=FDR

Methods for FDR control

The following FDR control methods are provided by the `stats::p.adjust` function

- Benjamini and Hochberg
- Benjamini and Yekutieli
- Q-value (provided by the Biocoductor `qvalue` package)

References

- Benjamini Y, Hochberg Y (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B*. 57: 289-300.
- Benjamini Y, Yekutieli D (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*. 29:1165-1188.
- Bonferroni CE (1936). Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*
- Holm S (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*. 6:65-70.
- Hochberg Y (1988). A sharper Bonferroni procedure for multiple tests of significance. *Biometrika*. 75:800-803.
- Hommel G (1988). A stagewise rejective multiple test procedure based on a modified Bonferroni test. *Biometrika*. 75:383-386.
- Storey JD (2003). The positive false discovery rate: A Bayesian interpretation and the q-value. *Annals of Statistics*. 31: 2013-2035.