Research Informatics Core

Advanced Statistics in EdgeR

February 21, 2023

Outline

- **Morning**
  - Differential expression for digital counts data
  - Basic principles and usage of edgeR
  - Pair-wise comparisons
  - One-way ANOVA
- **Afternoon**
  - Two-way ANOVA
  - Repeated measures models
  - Batch effects
  - Application in different –omics settings

Confirm you have these packages installed in your R studio:

edgeR
ComplexHeatmap
circlize (helps us with heatmap colors)
ggplot2
biomaRt (optional)

To check:

1) Open R studio
2) Run the commands:

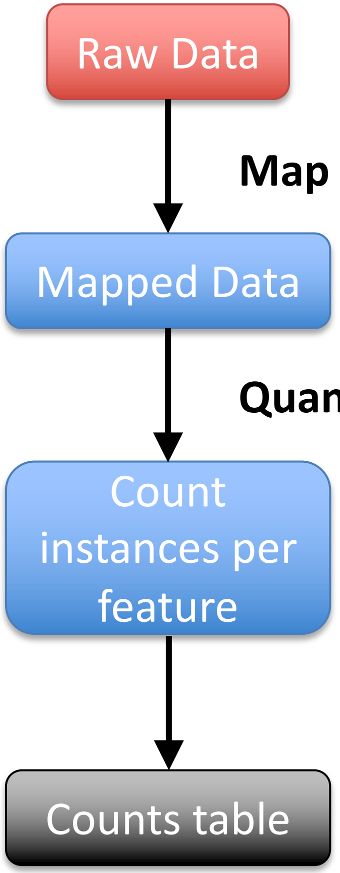
```
library(edgeR)
library(ComplexHeatmap)
library(circlize)
library(ggplot2)
```

(optional exercise)

```
library(biomaRt)
```

3) If no errors, then you're OK

Typical –omics data quantification


Raw Data

Mapped Data

Count instances per feature

Counts table

Map to reference database

Methods you might use…

STAR/Kallisto (RNA-seq); BWA MEM (ATAC-seq, ChIP-seq); Bismark (BS-seq); BLAST/Centrifuge (Shotgun Metagenomics); …

Quantify features

featureCounts/Kallisto (RNA-seq); Macs2/bedtools/featureCounts (ATAC-seq, ChIP-seq); Bismark (BS-seq); LCA (Shotgun Metagenomics); …

Features can include: *Genes (RNA-seq); Peaks (ATAC-seq, ChIP-seq), CpG (BS-seq); Taxonomic level (Metagenomics)*

Feature ID	A.1	A.2	B.1	B.2
ENSMUSG00000102948.1	1	0	1	0
ENSMUSG00000025902.13	0	0	1	2
ENSMUSG00000104238.1	0	0	0	0
ENSMUSG00000098104.1	57	37	69	69
ENSMUSG00000102175.1	5	4	1	0
ENSMUSG00000103922.1	11	6	5	16
ENSMUSG00000033845.13	758	790	657	609
ENSMUSG00000025903.14	743	871	1106	1020
ENSMUSG00000033813.15	569	605	576	486

Counts table:
Data are **digital** (integers), based on the **number of observations** of each feature in the data set. Scale of counts **depends on depth** of sequencing, **and breadth** of features in the data set.

Example pipeline overview: RNA-seq

1. For each sample:
   - Map reads to genome in splice-aware manner (STAR)
   - Quantify gene expression against gene annotation (featureCounts)
2. Combine counts from featureCounts across all samples
3. Prepare metadata table

Example pipeline overview: ATAC-seq

1. For each sample:
   - Map reads to genome (BWA MEM)
   - Remove PCR duplicates (Picard or Samtools)
   - Call peaks (Macs2)
2. Merge peak calls across all samples (bedtools merge), convert to "SAF" format for use with featureCounts
3. For each sample:
   - Quantify peak abundance against merged peaks (featureCounts)
4. Combine counts from featureCounts across all samples
5. Prepare metadata table

Example pipeline overview: Shotgun metagenomics (read-based annotation)

1. For each sample:
   - Map reads to taxonomic database (Centrifuge)
   - Summarize alignments for each read to least common ancestor (Megan)
   - Count alignments to taxon
   - Map reads to functional database in translated search (Diamond)
   - Summarize alignments to Kegg Ortholog (KO)
   - Count alignments per KO
2. Combine counts for taxa and functions across all samples
3. Prepare metadata table

Taxonomic quantification

Functional quantification

Exercise 1.1: Preparing input data

As we would for RNA-seq:

- Read featureCounts tables into R, combine, and save as tab-delimited text
- Prepare a metadata file in Excel and save as tab-delimited text

Analyzing a counts table for differential expression

- We need the counts and the metadata
- Filter features to analyze
  - Usually based on minimum abundance
- Normalization
  - Usually CPM – counts per million
- Differential expression
  - Fold-changes
  - P-values, Q-values (FDR correction)
  - Multi-group and multi-factor comparisons
- Exploration/visualization of differential stats
- **For today, we'll use "gene expression" for most of our examples, but all notes extend very generally to other –omics applications**

Counts table

Feature ID	A.1	A.2	B.1	B.2
ENSMUSG00000102948.1	1	0	1	0
ENSMUSG00000025902.13	0	0	1	2
ENSMUSG00000104238.1	0	0	0	0
ENSMUSG00000098104.1	57	37	69	69
ENSMUSG00000102175.1	5	4	1	0
ENSMUSG00000103922.1	11	6	5	16
ENSMUSG00000033845.13	758	790	657	609
ENSMUSG00000025903.14	743	871	1106	1020
ENSMUSG00000033813.15	569	605	576	486

Metadata table

Sample ID	Group
A.1	A
A.2	A
B.1	B
B.2	B

Notes on metadata table

- Best to store metadata in a separate table
  - Formatting sample names with metadata becomes awkward with a lot of factors
  - Do not rely on colors or special formatting in Excel to differentiate groups. Add a factor/column instead.
    - These are lost when exported to tabular, or imported in R, and you can’t sort or filter on them in Excel
- Pick simple sample names
  - Stick to alphanumeric (A-Z, 0-9), and underscore (_) and period (.)
  - **Avoid dash (-):** R turns these into periods for column names
  - Avoid spaces and other special characters (+/*&?%$@!<>;:’”)
- Reference guide:

https://github.com/uic-ric/uic-ric.github.io/wiki/Providing-sample-information

Normalization

Gene expression needs to be comparable between samples

Standard unit is **CPM** (counts per million), also sometimes TPM (transcripts per million), or RPM (reads per million).

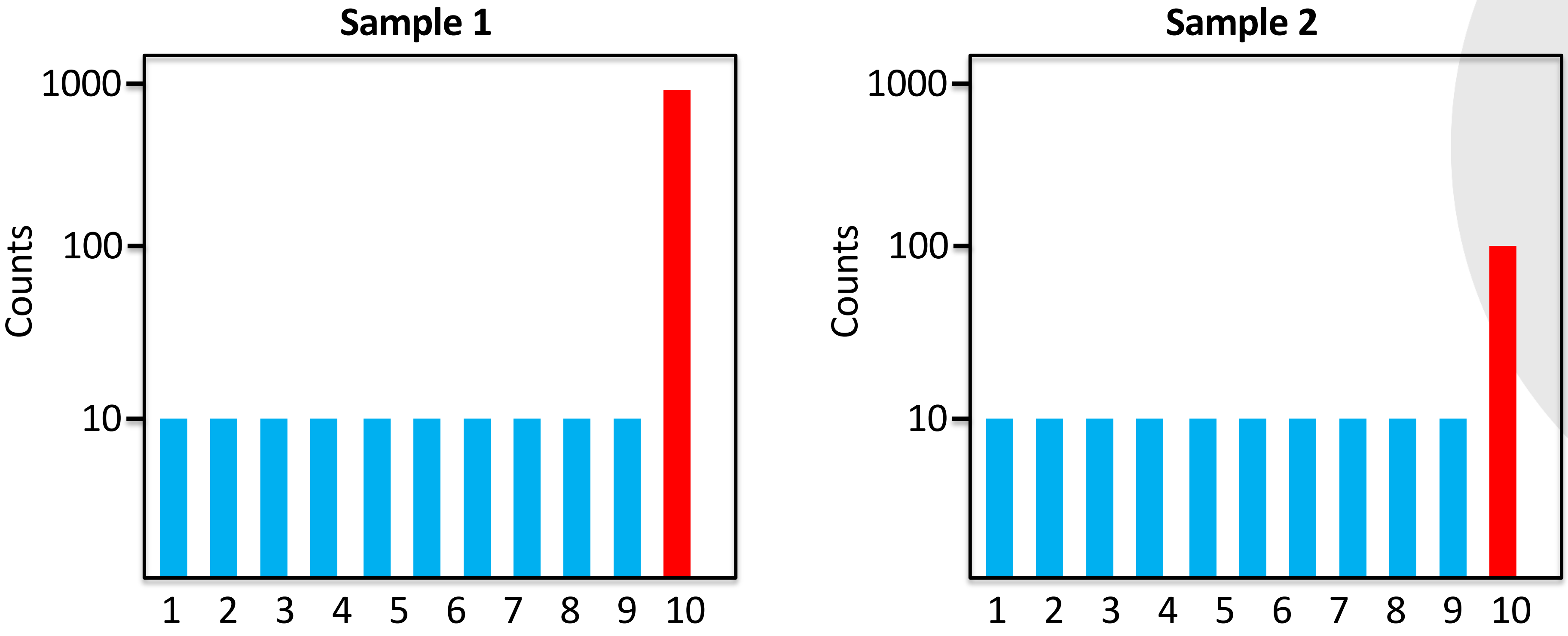
$$\text{CPM} = \frac{\text{Reads in transcript (feature)}}{\text{Millions of reads in sample}}$$

Equal to percent x 1 million

Additional normalization

- Sometimes CPM can be skewed if you have a few very highly expressed genes
  - CPM = percent x 1 million

Simple example: 10 genes

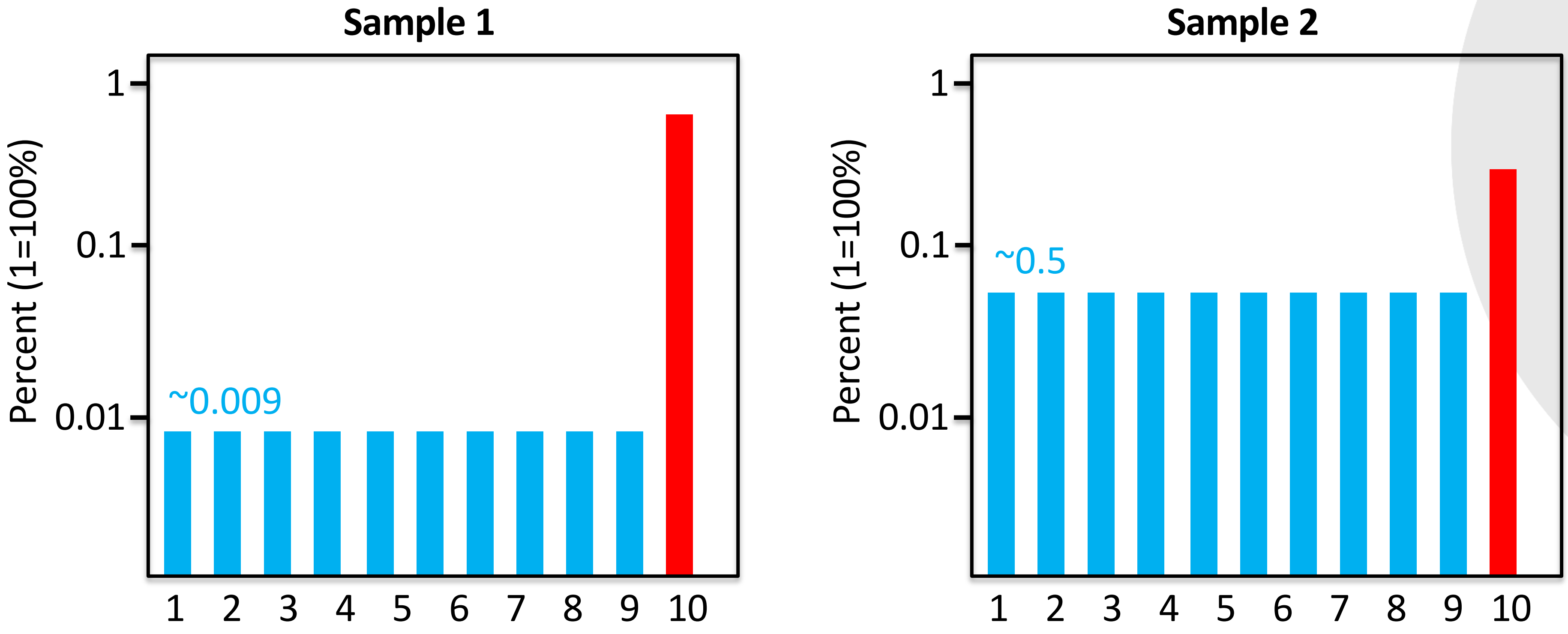


Raw counts

Additional normalization

- Sometimes CPM can be skewed if you have a few very highly expressed genes
  - CPM = percent x 1 million

Simple example: 10 genes

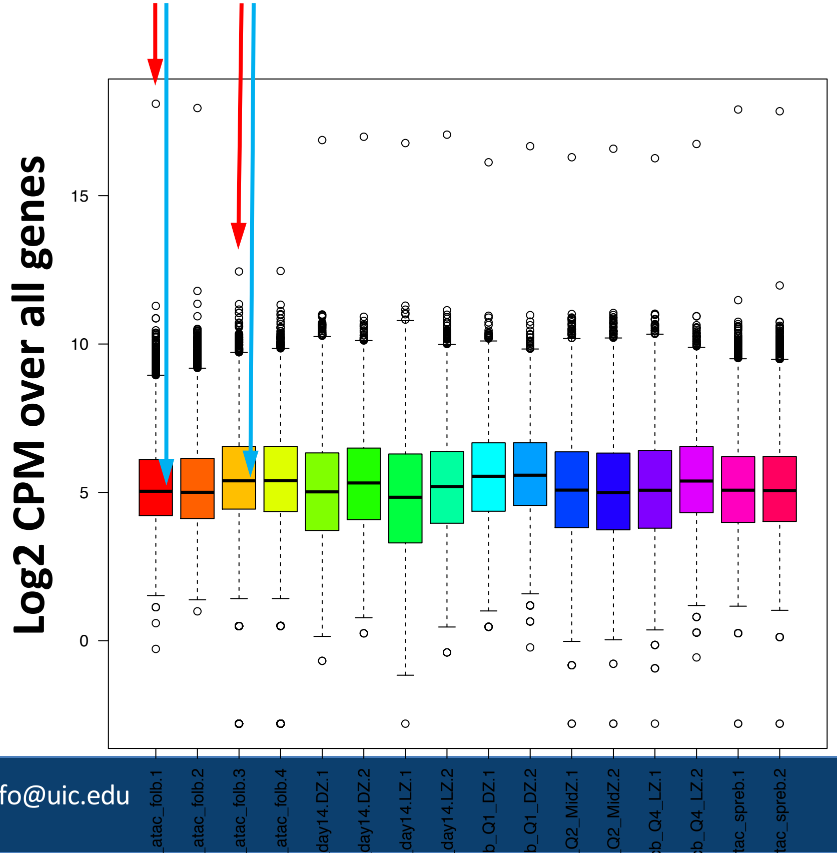


CPM normalized counts

Additional normalization

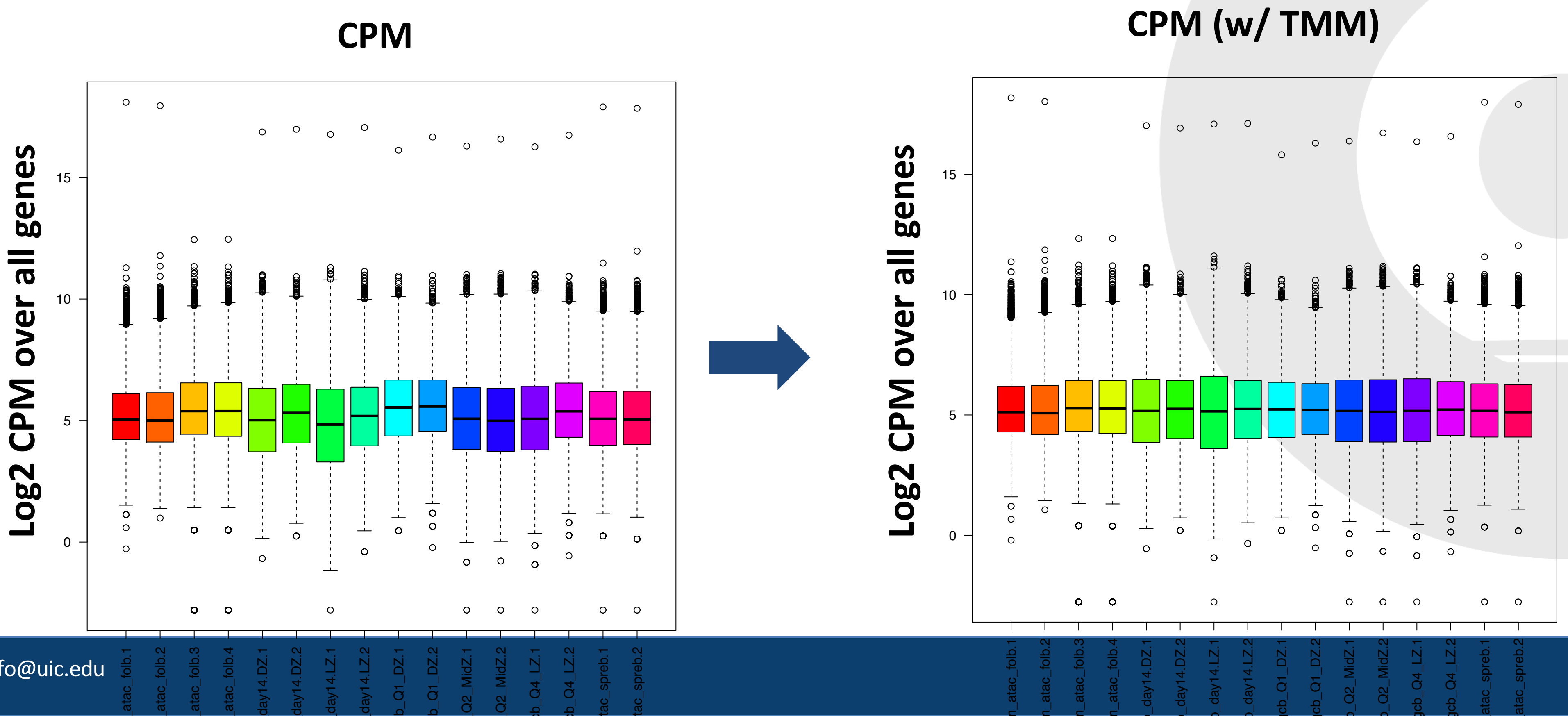
- CPM skew in real dataset

Higher expression of the top gene lowers the median normalized expression: a bigger fraction of reads are going to the top gene



Additional normalization

- TMM normalization (edgeR): calculate an extra factor under the assumption that most genes have log-fold-change of 0
  - Trimmed mean: Ignore the top and bottom few % of genes in calculation of normalization factor


CPM

CPM (w/ TMM)

Differential analysis

Compare within-group variability to between-group variability

- Biological replicates are essential (at least 3 recommended for model systems)

Similar idea to t-test or ANOVA, but:

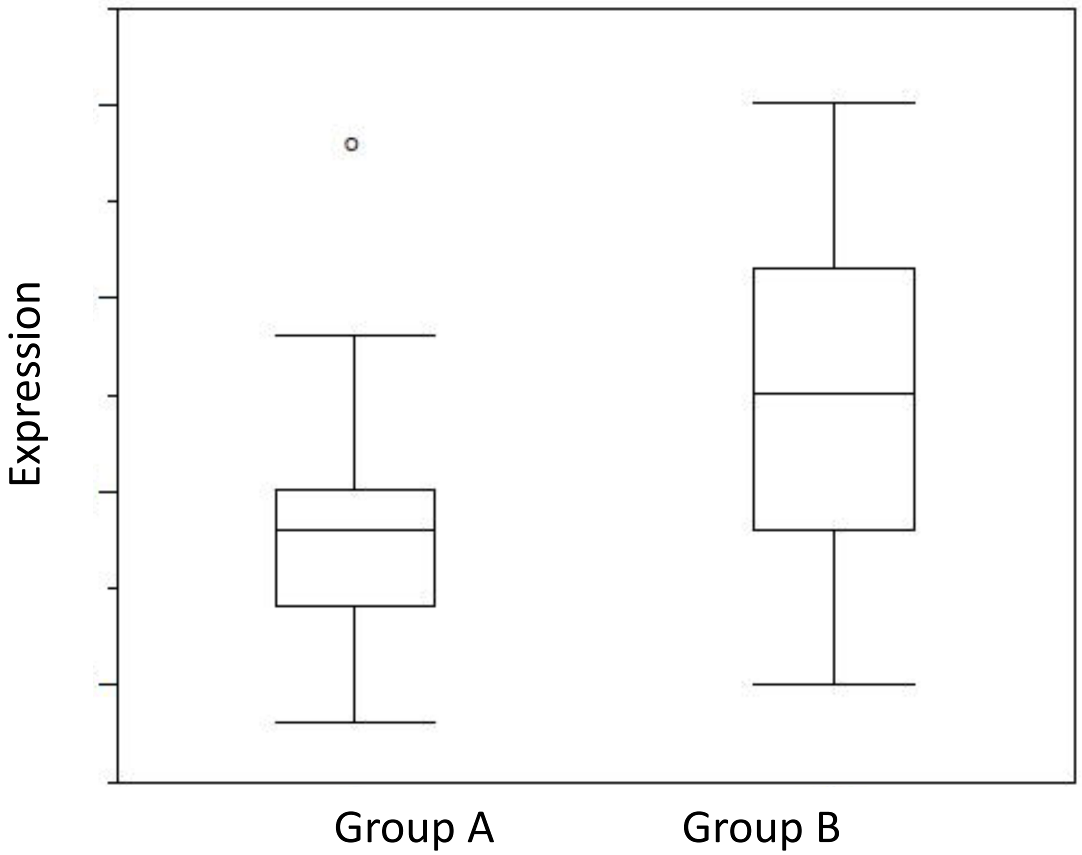
- Variance is non-normal
- Use negative binomial on raw counts

(t-test can't be used for data with negative binomial distribution)

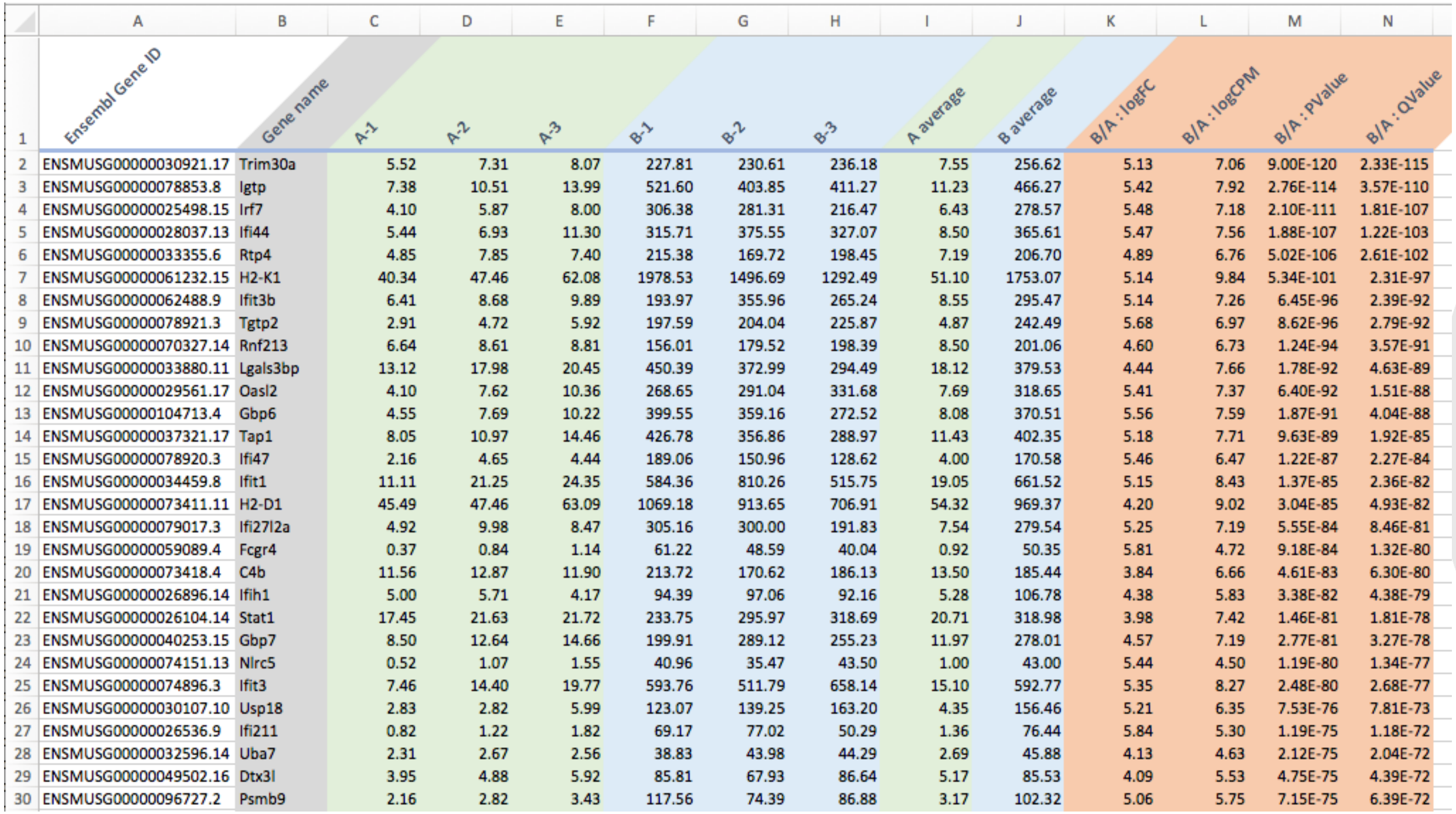
R packages:

- EdgeR
- DESeq2

Run FDR (a.k.a Benjamini-Hochberg) correction to adjust for multiple testing



Example results



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Ensembl Gene ID	Gene name	A-1	A-2	A-3	B-1	B-2	B-3	A average	B average	B/A : logFC	B/A : logCPM	B/A : PValue	B/A : QValue
2	ENSMUSG00000030921.17	Trim30a	5.52	7.31	8.07	227.81	230.61	236.18	7.55	256.62	5.13	7.06	9.00E-120	2.33E-115
3	ENSMUSG00000078853.8	Igtp	7.38	10.51	13.99	521.60	403.85	411.27	11.23	466.27	5.42	7.92	2.76E-114	3.57E-110
4	ENSMUSG00000025498.15	Irf7	4.10	5.87	8.00	306.38	281.31	216.47	6.43	278.57	5.48	7.18	2.10E-111	1.81E-107
5	ENSMUSG00000028037.13	Ifi44	5.44	6.93	11.30	315.71	375.55	327.07	8.50	365.61	5.47	7.56	1.88E-107	1.22E-103
6	ENSMUSG00000033355.6	Rtp4	4.85	7.85	7.40	215.38	169.72	198.45	7.19	206.70	4.89	6.76	5.02E-106	2.61E-102
7	ENSMUSG00000061232.15	H2-K1	40.34	47.46	62.08	1978.53	1496.69	1292.49	51.10	1753.07	5.14	9.84	5.34E-101	2.31E-97
8	ENSMUSG00000062488.9	Ifit3b	6.41	8.68	9.89	193.97	355.96	265.24	8.55	295.47	5.14	7.26	6.45E-96	2.39E-92
9	ENSMUSG00000078921.3	Tgtp2	2.91	4.72	5.92	197.59	204.04	225.87	4.87	242.49	5.68	6.97	8.62E-96	2.79E-92
10	ENSMUSG00000070327.14	Rnf213	6.64	8.61	8.81	156.01	179.52	198.39	8.50	201.06	4.60	6.73	1.24E-94	3.57E-91
11	ENSMUSG00000033880.11	Lgals3bp	13.12	17.98	20.45	450.39	372.99	294.49	18.12	379.53	4.44	7.66	1.78E-92	4.63E-89
12	ENSMUSG00000029561.17	Oasl2	4.10	7.62	10.36	268.65	291.04	331.68	7.69	318.65	5.41	7.37	6.40E-92	1.51E-88
13	ENSMUSG00000104713.4	Gbp6	4.55	7.69	10.22	399.55	359.16	272.52	8.08	370.51	5.56	7.59	1.87E-91	4.04E-88
14	ENSMUSG00000037321.17	Tap1	8.05	10.97	14.46	426.78	356.86	288.97	11.43	402.35	5.18	7.71	9.63E-89	1.92E-85
15	ENSMUSG00000078920.3	Ifi47	2.16	4.65	4.44	189.06	150.96	128.62	4.00	170.58	5.46	6.47	1.22E-87	2.27E-84
16	ENSMUSG00000034459.8	Ifit1	11.11	21.25	24.35	584.36	810.26	515.75	19.05	661.52	5.15	8.43	1.37E-85	2.36E-82
17	ENSMUSG00000073411.11	H2-D1	45.49	47.46	63.09	1069.18	913.65	706.91	54.32	969.37	4.20	9.02	3.04E-85	4.93E-82
18	ENSMUSG00000079017.3	Ifi27l2a	4.92	9.98	8.47	305.16	300.00	191.83	7.54	279.54	5.25	7.19	5.55E-84	8.46E-81
19	ENSMUSG00000059089.4	Fcgr4	0.37	0.84	1.14	61.22	48.59	40.04	0.92	50.35	5.81	4.72	9.18E-84	1.32E-80
20	ENSMUSG00000073418.4	C4b	11.56	12.87	11.90	213.72	170.62	186.13	13.50	185.44	3.84	6.66	4.61E-83	6.30E-80
21	ENSMUSG00000026896.14	Ifih1	5.00	5.71	4.17	94.39	97.06	92.16	5.28	106.78	4.38	5.83	3.38E-82	4.38E-79
22	ENSMUSG00000026104.14	Stat1	17.45	21.63	21.72	233.75	295.97	318.69	20.71	318.98	3.98	7.42	1.46E-81	1.81E-78
23	ENSMUSG00000040253.15	Gbp7	8.50	12.64	14.66	199.91	289.12	255.23	11.97	278.01	4.57	7.19	2.77E-81	3.27E-78
24	ENSMUSG00000074151.13	Nlrc5	0.52	1.07	1.55	40.96	35.47	43.50	1.00	43.00	5.44	4.50	1.19E-80	1.34E-77
25	ENSMUSG00000074896.3	Ifit3	7.46	14.40	19.77	593.76	511.79	658.14	15.10	592.77	5.35	8.27	2.48E-80	2.68E-77
26	ENSMUSG00000030107.10	Usp18	2.83	2.82	5.99	123.07	139.25	163.20	4.35	156.46	5.21	6.35	7.53E-76	7.81E-73
27	ENSMUSG00000026536.9	Ifi211	0.82	1.22	1.82	69.17	77.02	50.29	1.36	76.44	5.84	5.30	1.19E-75	1.18E-72
28	ENSMUSG00000032596.14	Uba7	2.31	2.67	2.56	38.83	43.98	44.29	2.69	45.88	4.13	4.63	2.12E-75	2.04E-72
29	ENSMUSG00000049502.16	Dtx3l	3.95	4.88	5.92	85.81	67.93	86.64	5.17	85.53	4.09	5.53	4.75E-75	4.39E-72
30	ENSMUSG00000096727.2	Psmb9	2.16	2.82	3.43	117.56	74.39	86.88	3.17	102.32	5.06	5.75	7.15E-75	6.39E-72

Gene annotation | Expression per sample (CPM) | Averages | Differential stats

Typical scales for mouse/human:

RNA-seq:

~40,000 genes

ATAC-seq:

~40,000-150,000 peaks

ChIP-seq:

~10,000-60,000 peaks

BS-seq (RRBS):

~1,000,000 CpGs

***Metagenomics*:**

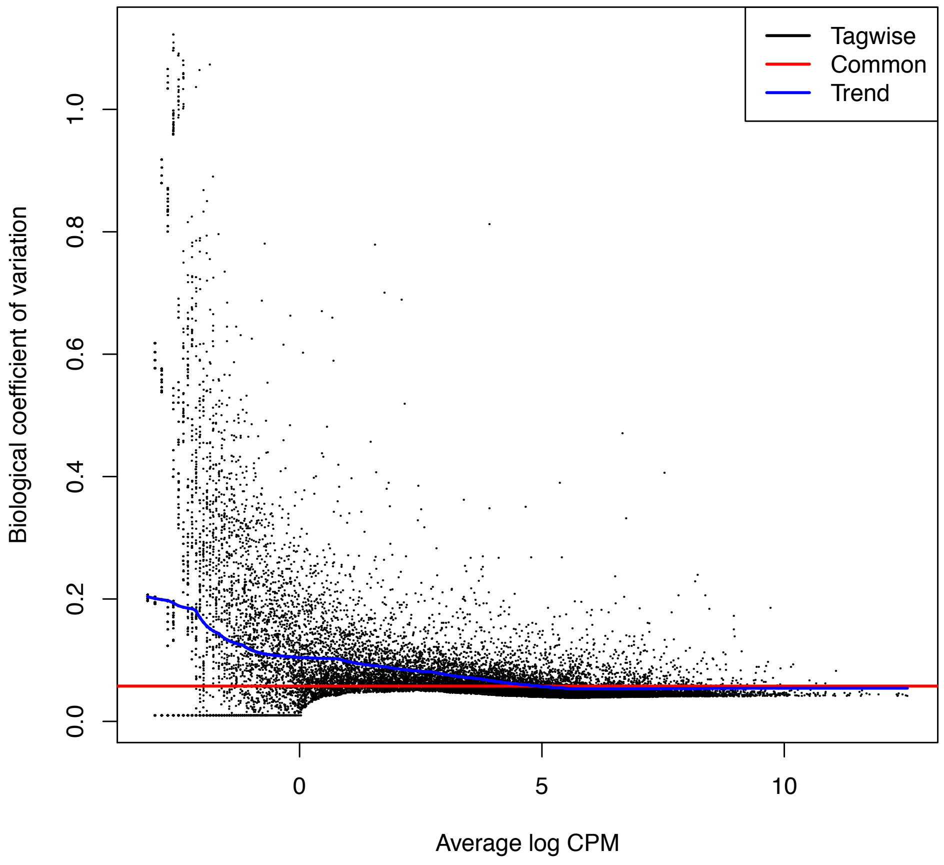
~10-100 phlya; ~500-5000 species

edgeR

- R package for normalization and differential analysis
  - Similar to DESeq2
- Computes CPM normalization, TMM norm factors
- Pair-wise differentials (“exactTest”)
- Generalized linear models (GLM) for multi-group and multi-factors differential statistics
  - 2-factor example:
    - WT/Mut (factor 1: genotype)
    - Control/treated (factor 2: treatment)
  - 2-factor analysis similar to a 2-way ANOVA:
    - Test effect of genotype, independent on treatment
    - Test effect of treatment, independent of genotype
    - Test for interaction between genotype and treatment
  - Allows us to correct for batch effects

Dispersion

- Conceptually similar to variance
  - **Common** dispersion: general variance in the entire data set
  - **Trended** dispersion: trend of dispersion as a function of gene expression level
  - **Tagwise** dispersion: dispersion per gene
- BCV (biological coefficient of variation) = square-root dispersion
- Typical BCV values:
  - Technical replicates: 0.01 (i.e., 1%)
  - Model organisms: 0.1-0.3 (10%-30%)
  - Human cohorts: 0.5 (50%) or higher
- If you don't have replicates: you have to guess a common dispersion and go with that (very much not recommended)



Pairwise Differential Expression

- **exactTest()** in edgeR:

  *Given **estimates** for the dispersion and normalization factors (TMM), compute exact p-values by **summing all probabilities** in the negative binomial distribution that are less than the observed probability of the observed counts*

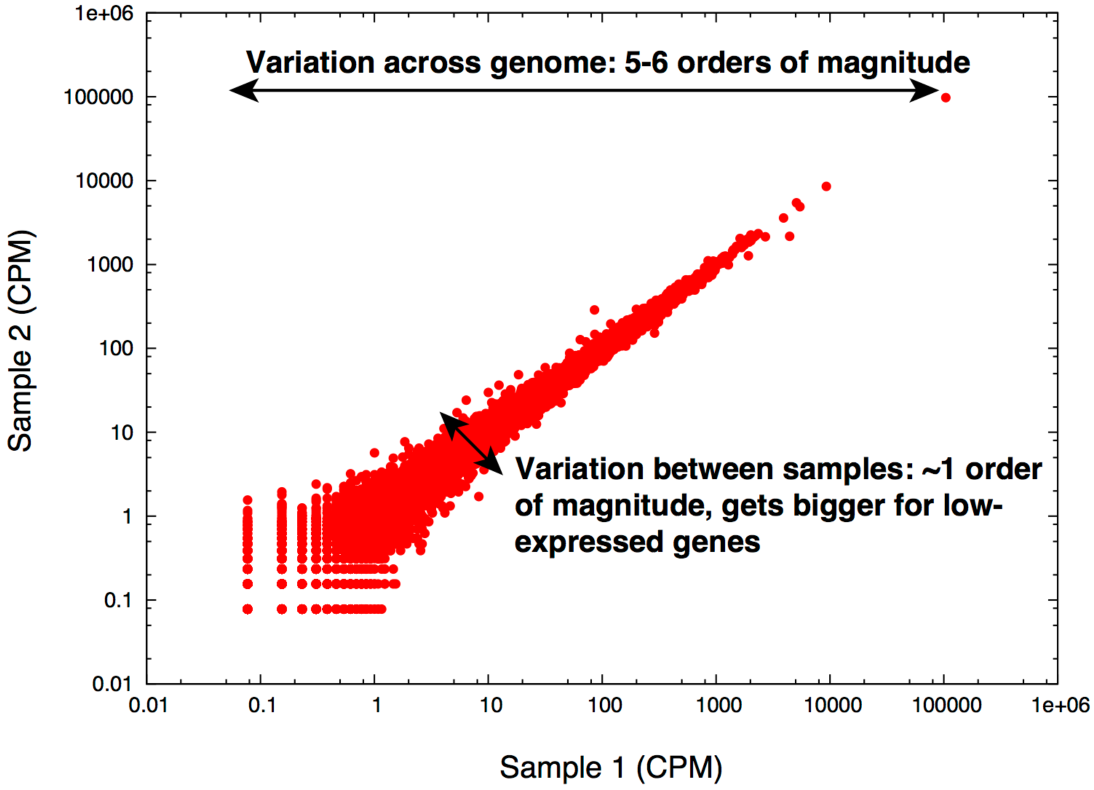
- This test only works for a single factor, though may not scale well for more than a pair-wise comparison
- Not really “exact”, as it is relying on *estimates* of, e.g., the dispersion
  - The magnitude of the dispersion has a huge effect on your power

Exercise 1.2: edgeR

- Read data into R
- Set up edgeR objects
- Compute normalization factors
- Estimate dispersion
- Run differential analysis, and FDR correction
- Generate normalized expression table
- Write out results

Downstream analysis: data transformations

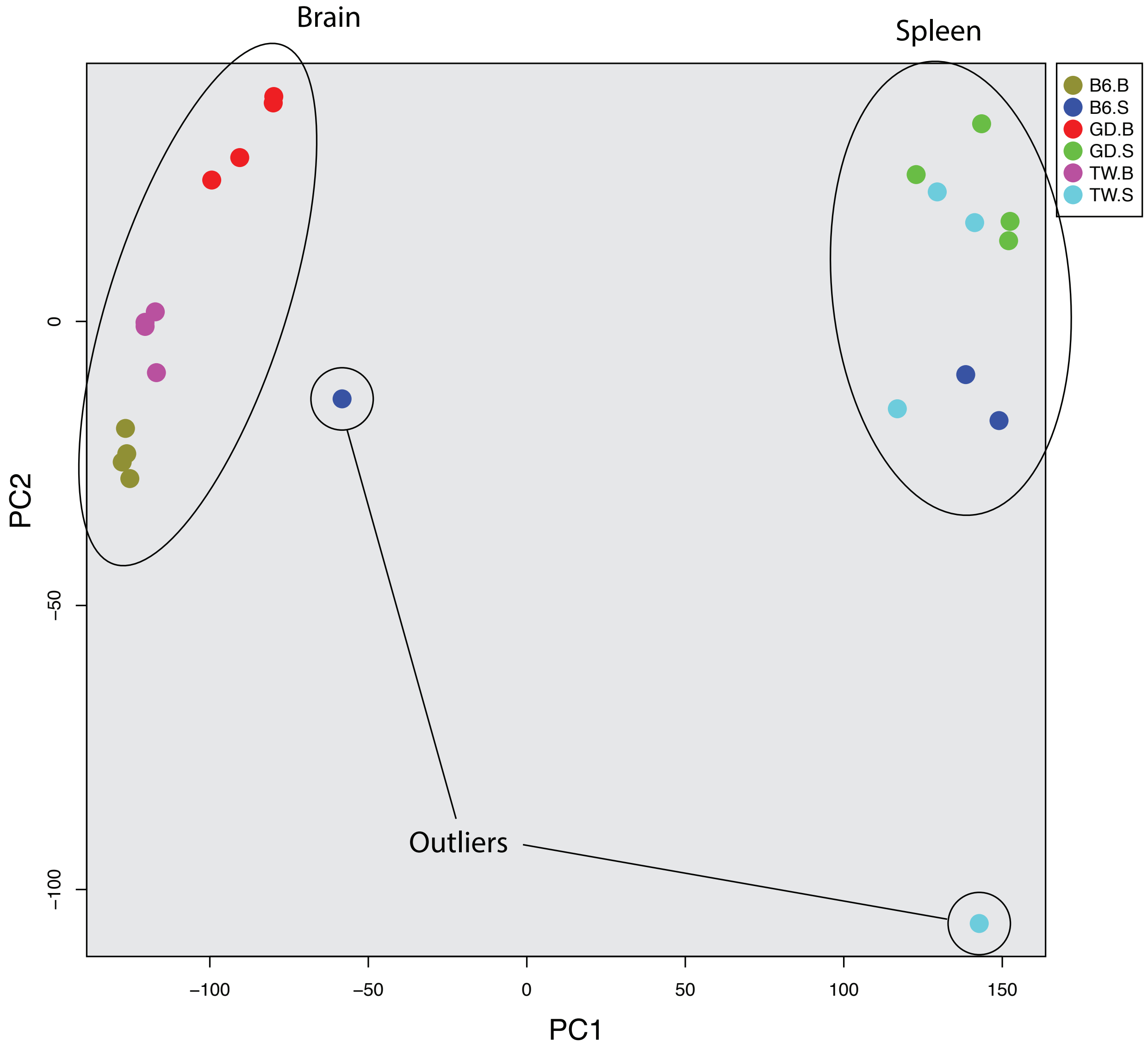
- For high-level visualizations, we want to **emphasize relative changes between samples**
- Log-scale (usually log2)
  - Symmetrize increase/decrease (2-fold change is +/-1 in log2)
  - Make stats more uniform across wide range of expression levels
- Z-score
  - Z = (x-mean)/SD
  - Controls for differences in expression level and variability
  - Useful for clustering, plotting data in heatmaps



High-level clustering and QC with PCA

Principal component analysis (PCA)

- Orthogonal transformation normalized expression levels covariation matrix
- Dimensionality reduction
- Quick visualization of sample similarity
- Batch effects
- Outliers

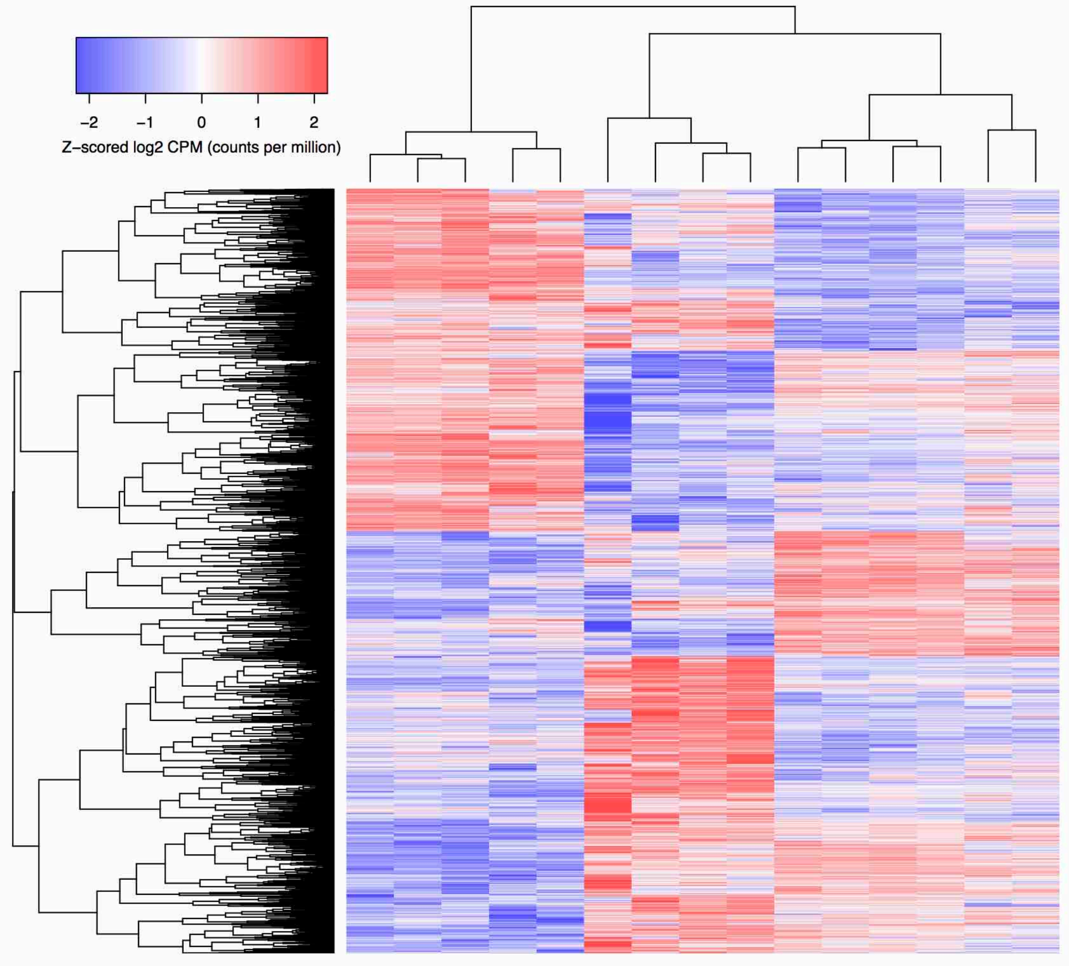


Exercise 1.3: PCA plot

- Run PCA on log-scaled normalized expression
  - (We could have done this in edgeR: set “log=T” in the cpm function; edgeR adds a pseudo-count to avoid taking the log of 0)
  - We need to transpose the table to run PCA on the *samples*
- Plot PC1 vs PC2
- Check the % variance for each PC (summary and screeplot)

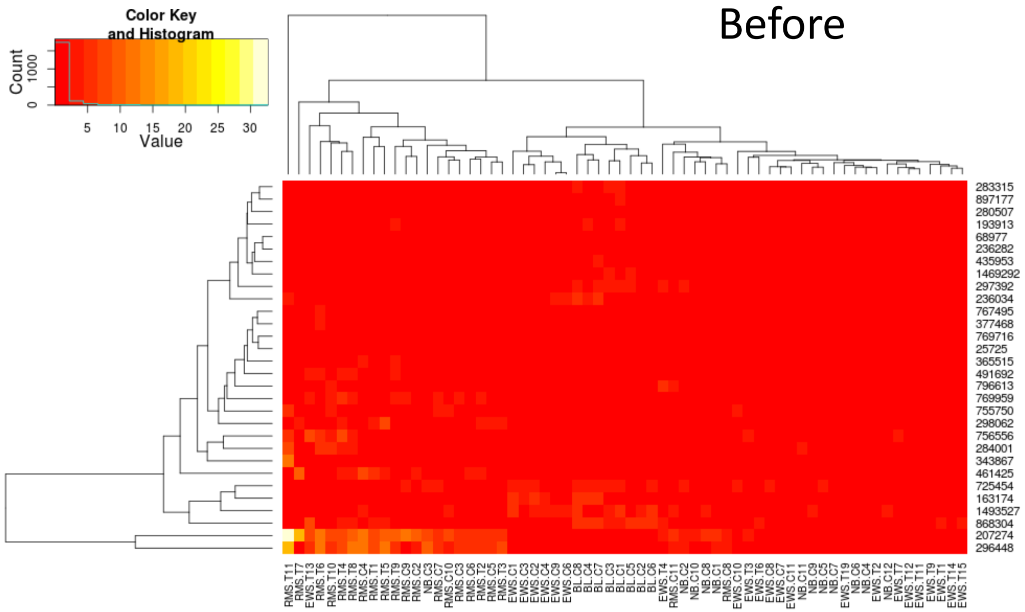
Visualization of Gene Expression

- Heatmap
  - Graphical representation of the values in a matrix
  - Each row represents a gene
  - Each column represent a sample
  - Each color box in heatmap represents the expression level of a gene in a certain sample
- For >2 groups, we often turn to clustering to find transcriptional patterns
  - Perform initial filtering using multi-group significance test (similar to ANOVA)
  - Log-scale and Z-score normalized expression levels
- Hierarchical clustering + heatmaps



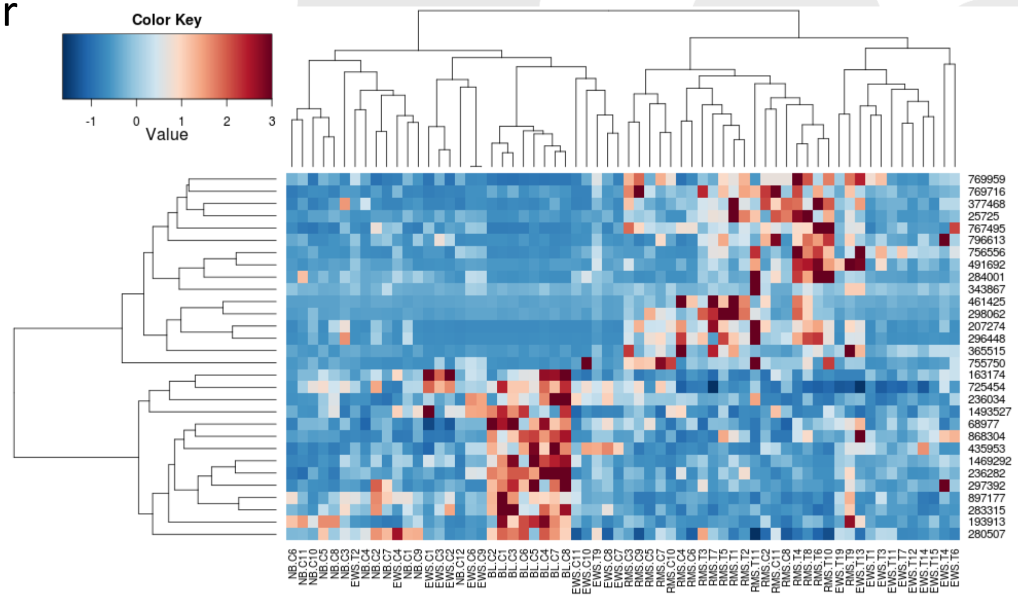
Heatmap – effect of scaling

- Major concerns
  - Expression changes across sample groups may be much smaller than gene-to-gene differences *within* a sample. But we mainly care about changes across groups
  - Z-score normalization
    - Visualize the gene expression on the same baseline
    - Pop out the pattern of differential expression



Before

After



Exercise 1.4: Heatmap

- Make a simple heatmap of the differentially expressed genes
- First: we need to get the log-scaled normalized expression for *just* the differentially expressed genes, and then z-score it
- Plot heatmaps with the Heatmap function in ComplexHeatmap

Aside: RNA-seq power calculations

- Web tool: https://cqs-vumc.shinyapps.io/rnaseqsamplesizeweb/
- Parameters:
  - Significance level (FDR)
  - Normalization factors
  - Total genes
  - Expected significant genes
  - Minimum fold-change
  - Read depth for significant genes
  - **Dispersion** ← Most important parameter, has the biggest effect on power, and the one you probably know the least about

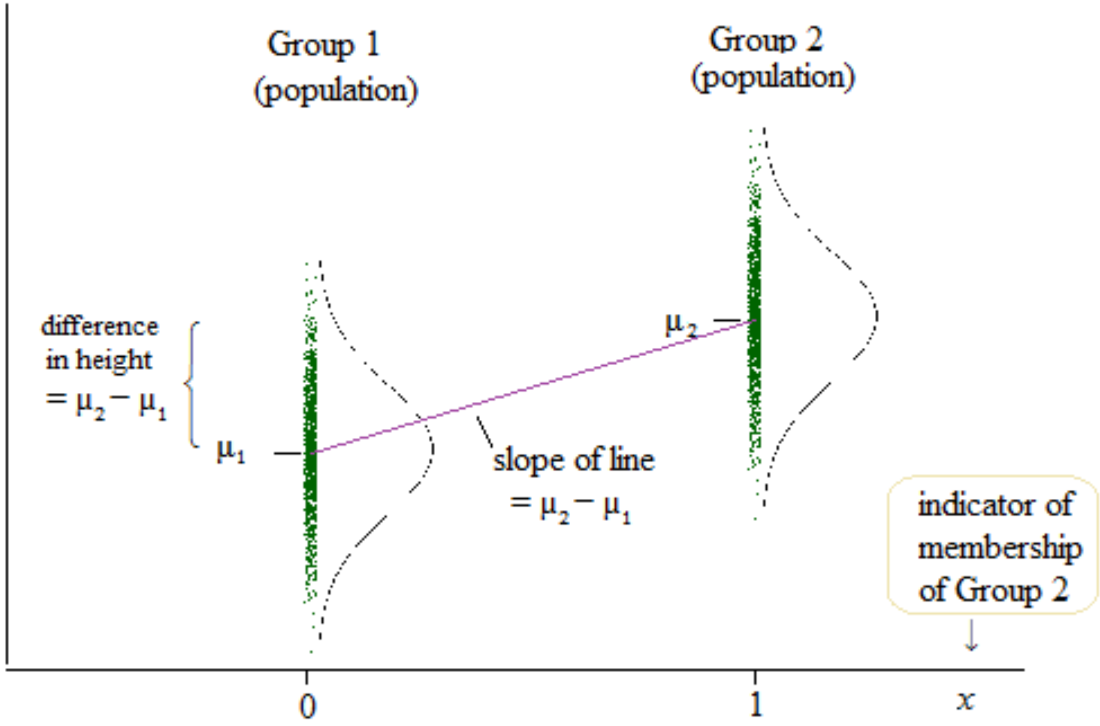
BREAK

Multi-group analysis (ANOVA)

- How to analyze across multiple groups?
  - Number of **factors**
    - Genotype, Treatment, Time point, Sex (4 factors)
  - Number of **levels** per factor
    - Genotype: WT, Gene A KO (2 levels)
    - Time point: time 1, time 2, time 3 (3 levels)
- One factor, 2 levels: pair-wise comparison (done)
- **One factor, >2 levels: one-way ANOVA (this morning)**
- Two factors: two-way ANOVA (this afternoon)
- Three factors: three-way ANOVA, etc.

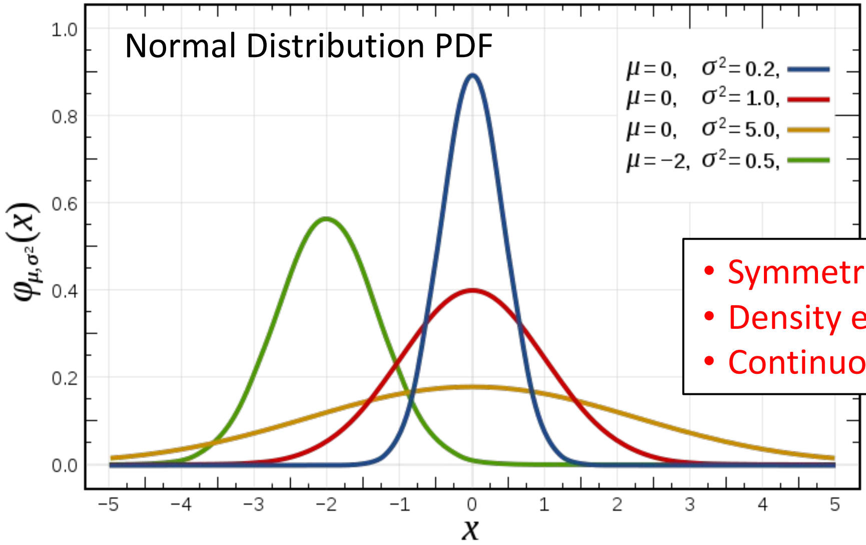
Link between ANOVA and multiple regression

- ANOVA and linear regression are essentially the same thing
  - ANOVA: Testing for difference in means across multiple groups
  - Linear Regression: Model the measurement (dependent variable) as a combination of groups
- In R, to run an ANOVA you first fit a linear model to the data
  - Fits and p-values are calculated under an assumption that the data are normally distributed

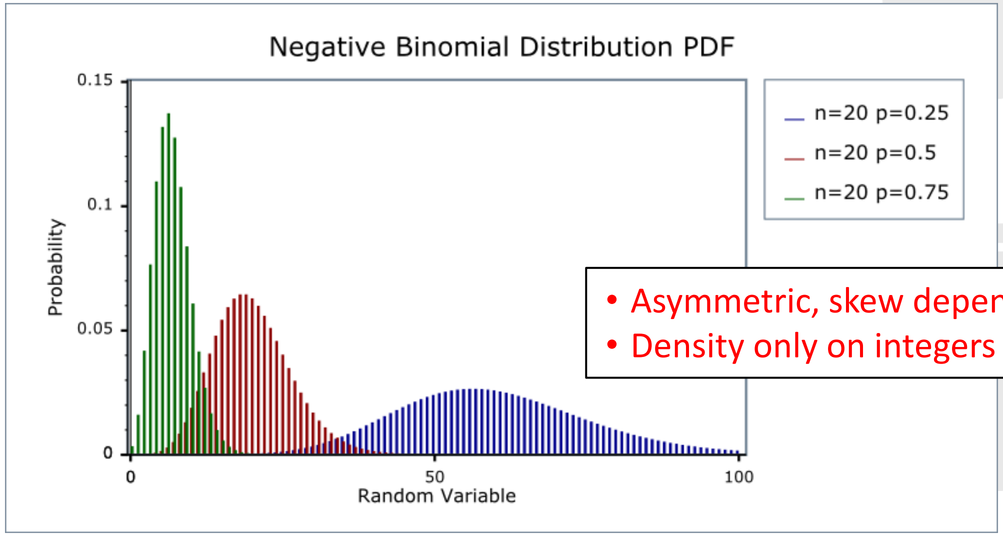


Multiple regression in edgeR: Generalized Linear Models (GLM)

- Normal distribution not appropriate for digital counts data
- Use negative binomial distribution
  - Density on integers
  - Minimum at 0
  - Unlike Poisson, variance can be set independently of the mean
    - AKA “Overdispersed Poisson”
- Generalized Linear Models (GLM): extension of linear models with alternative variance model



- Symmetric around mean
- Density extends to +/- infinity
- Continuous (covers all real numbers)



- Asymmetric, skew depends on the mean
- Density only on integers ≥ 0

Generating a linear model in R using model.matrix

Example data set with 1 factor, 3 levels

Metadata:

Sample ID	Treatment
A.1	A
A.2	A
B.1	B
B.2	B
C.1	C
C.2	C

Conceptual goal

Model gene expression as a function of treatment level (A/B/C)

High-level setup in R

```
treatment <- factor(c('A','A','B','B',
    'C','C'))
model.matrix(~ treatment)
```

Low-level model matrix in R

```
  (Intercept) treatmentB treatmentC
1           1          0          0
2           1          0          0
3           1          1          0
4           1          1          0
5           1          0          1
6           1          0          1
```

- "Intercept" effectively models treatment A as the baseline
- Other terms are modeling the deviation from A due to treatment B or C
- Our ANOVA test will look to see if these other terms have a significant impact in the model's accuracy

Equivalent linear equation

$y = \beta_0 + \beta_1*\text{treatmentB} + \beta_2*\text{treatmentC} + \varepsilon$

- y is expression level
- β's are fitted coefficients (β_0 for intercept). Dummy values of 0/1 based on grouping of each sample.
- ε is error

Testing for significance: Quasi-likelihood F-test

- Evaluate fitted coefficients on model accuracy:

Complete linear model:

$y = \beta_0 + \beta_1$*treatmentB + β_2*treatmentC + ε

  - Calculate the error (ε) in our full model
  - Drop terms of interest from the model, recalculate error

Alternative model:

$y = \beta_0 + \varepsilon$

- Formally, errors are recast as a model (quasi) *likelihood*
  - Quasi: allowing for overdispersion of model
  - P-values calculated from an F-test based on the **ratios of the likelihoods** for the two models
- We fit the model once, but can perform multiple tests by dropping or comparing different combinations of terms in the model.

Exercise 1.5: Set up and run one-way ANOVA-type model

- Basic setup for GLM processing in edgeR
- Dispersion estimation and DE stats
- Make a heatmap

NOTE: messages from Heatmap() about rasterizing the heatmap can be ignored. Rasterizing (making into bitmaps) the heatmaps makes the saved images much smaller, but lower resolution, for really big heatmaps. You can control this behavior directly if you want with the `use_raster = T/F` ***parameter.***

Contrasts

- Test more specific questions within a GLM setup
- Easier to construct model matrix without intercept
  - Start the formula with “0+” to indicate no intercept
  - Each level has its own term

```
model2 <- model.matrix(~0+treatment)
> model2
   treatmentA treatmentB treatmentC
1           1          0          0
2           1          0          0
3           0          1          0
4           0          1          0
5           0          0          1
6           0          0          1
```

What we had before

```
  (Intercept) treatmentB treatmentC
1           1          0          0
2           1          0          0
3           1          1          0
4           1          1          0
5           1          0          1
6           1          0          1
```

Setting up contrasts

- Define which specific groups we want to compare
  - **makeContrasts** function lets you set up comparisons between groups in different ways

```
comp <- makeContrasts(treatmentA - treatmentB, levels=model)
```

Directly compare A to B (i.e., pairwise), without considering treatment C

** But treatment C's effect is still implicit in the calculation, as it was included in the dispersion and TMM calculations, and in fitting the model*

-OR-

```
comp <- makeContrasts(treatmentC - (treatmentA+treatmentB)/2, levels=model)
```

Compare C to the average A and B

  - Then to test:

```
glmQLFTest(fit, contrast=comp)
```

  - No need to rerun dispersions and fit between comparisons

Exercise 1.6: Using contrasts

- Use contrasts in our GLM – 2 ways
- Plot heatmaps of differentially expressed genes

To use the intercept or not…

- There isn't a right or wrong answer for including the intercept
  - Include intercept (default behavior): `model.matrix(~treatment)`
  - Don't include intercept: `model.matrix(~0+treatment)`
  - Dispersions are the same (BCV 17% either way in last exercise). **The models are ultimately the same too!**
- We could also do the ANOVA-style test using the no-intercept model with this contrast: control vs average of all groups

```
comp <- makeContrasts(treatControl - (treatControl + treatModel1 + treatModel2)/3,
   levels=model)
```

↑ (This factor could be any level, doesn't *need* to be control)

  - This is awkward to write out, especially if you have a lot of groups

To use the intercept or not…

- Ultimately, pick the most convenient one
  - An ANOVA-style test is easier to set up with the intercept
  - Direct comparisons between combinations of groups (contrasts) are easier without the intercept
  - **YOU** can decide based on your question

Additional steps

- You will often want to follow up a multi-group comparison with different pair-wise comparisons
  - To test for overall significance, look at the multi-group test
  - To determine which groups are changing, look to specific pair-wise comparisons or contrasts
- Two options for pair-wise comparisons:
  1. Run a series of contrasts with GLM model
     - Faster to run, as does not require re-estimation of dispersion and GLM fit
  2. Run a series of pairwise comparisons with exactTest
     - More sensitive if different groups have different variances

Final thoughts for morning

- GLM capabilities in edgeR let us address a huge range of questions
  - This morning: 1 factor models
  - This afternoon: 2+ factor models
  - YOU have to determine what the right question is
- It's often useful to plot heatmaps of differentially expressed genes to get a sense of what's going on
- Get used to using loops and other programming structures in R
  - E.g., for running all pairwise comparisons

Exercise 1.7: Run all pairwise comparisons with exactTest

- Write a loop in R to run all of the pairwise comparisons
- We can also write this as a function for easy reuse on other data sets (time permitting)

LUNCH

Afternoon

More than one factor

- With more factors, there are more possible questions
  - Factors: Genotype, Treatment, Time point, Sex, …
  - What is affected by genotype?
  - What is affected by treatment?
  - What is affected by both?
  - What is affected by both, but in different ways?
- Do I want to model this as multiple factors, or combine the factors into a one-factor study?

Thinking about our metadata

- **Plan to list each factor in its own column**
- We will look at 2-factor models this afternoon
- Easy to extend to more than 2 factors

Metadata table: 2 factors

Sample ID	Genotype	Treatment
A.1	WT	Control
A.2	WT	Control
B.1	WT	Drug
B.2	WT	Drug
C.1	KO	Control
C.2	KO	Control
D.1	KO	Drug
D.2	KO	Drug

Metadata table: 3 factors

Sample ID	Genotype	Treatment	Sex
A.1	WT	Control	M
A.2	WT	Control	M
B.1	WT	Drug	M
B.2	WT	Drug	M
C.1	KO	Control	M
C.2	KO	Control	M
D.1	KO	Drug	M
D.2	KO	Drug	M
E.1	WT	Control	F
E.2	WT	Control	F
F.1	WT	Drug	F
F.2	WT	Drug	F
G.1	KO	Control	F
G.2	KO	Control	F
H.1	KO	Drug	F
H.2	KO	Drug	F

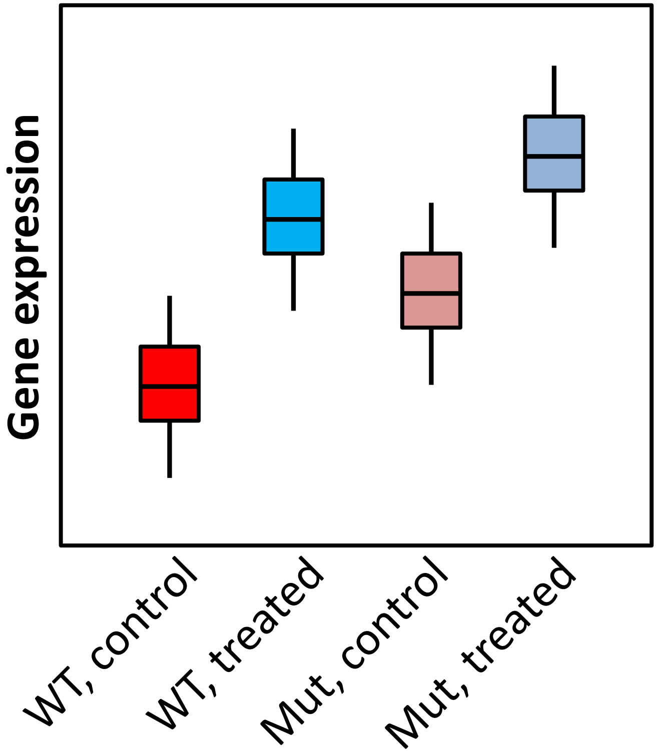
Modeling factors independently

- Model the effects of each factor independently of the other
  - Effect of genotype while controlling for treatment
  - Effect of treatment while controlling for genotype
  - Get 2 p-values and q-values
- Control for effects we don’t care about, but might make a difference: more explanatory power
  - Sex (may have an effect, not not primary outcome of interest)
  - Batch effect (later today)
  - Patient differences/repeated measures (later today)

Modeling combinatorial effects: interaction terms

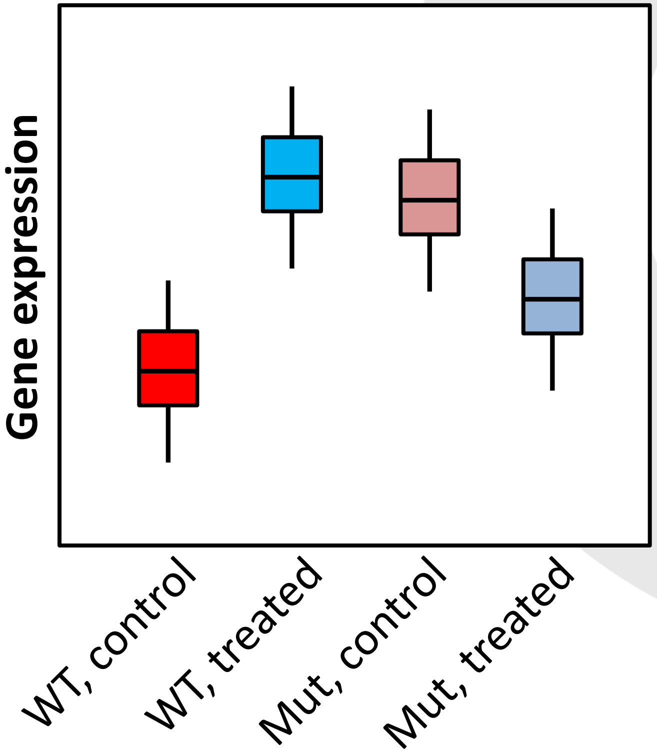
- Effect of one factor depends on the *level* of another factor
  - Treatment effect depends on genotype: treatment makes the gene go up in WT, but go down in mutant

Example 1: no interaction



Expression increases with treatment and in mutant, effects are additive. **Treatment and genotype effects are independent.**

Example 2: interaction



Expression increases with treatment for WT, but decreases for Mut. **Effect of treatment depends on genotype.**

GLM with multiple factors

Example data set with 2 factors

Metadata:

Sample ID	Genotype	Treatment
A.1	WT	Control
A.2	WT	Control
B.1	WT	Drug
B.2	WT	Drug
C.1	KO	Control
C.2	KO	Control
D.1	KO	Drug
D.2	KO	Drug

Conceptual goal

Model gene expression as a function of genotype and treatment

High-level setup in R

```
geno <- factor(c('WT','WT','WT','WT',
    'KO','KO','KO','KO'))
treat <- factor(c('C','C','D','D',
    'C','C','D','D'))
model.matrix(~ geno + treat + geno:treat)
```

Low-level model matrix in R

```
  (Intercept) genoWT treatD genoWT:treatD
1           1      1      0             0
2           1      1      0             0
3           1      1      1             1
4           1      1      1             1
5           1      0      0             0
6           1      0      0             0
7           1      0      1             0
8           1      0      1             0
```

- 1 variable for each factor (1 – number of levels)
- 1 variable for interaction – genoWT:treatD

Equivalent linear equation

$$y = \beta_0 + \beta_1 \text{*genoWT} + \beta_2 \text{*treatD} + \beta_3 \text{*(genoWT\_and\_treatD)} + \varepsilon$$

- y is expression level
- β’s are fitted coefficients (β_0 for intercept). Dummy values of 0/1 based on grouping of each sample.
- **β_3 captures extra effect due to combination of WT and Drug treatment**
- ε is error

Complete design

- To model as 2 factors, you need some samples in all groups
- To model interaction terms, you need some samples in all group *combinations*

Complete design

Sample ID	Genotype	Treatment
A.1	WT	Control
A.2	WT	Control
B.1	WT	Drug
B.2	WT	Drug
C.1	KO	Control
C.2	KO	Control
D.1	KO	Drug
D.2	KO	Drug

edgeR errors from incomplete design:

```
Design matrix not of full rank.  The
following coefficients not estimable...
```

Incomplete design 1

Sample ID	Genotype	Treatment
A.1	WT	Control
A.2	WT	Control
D.1	KO	Drug
D.2	KO	Drug

Can’t do any two-way test. Statistically, we can’t tell the difference between KO effect and drug effect

Incomplete design 2

Sample ID	Genotype	Treatment
A.1	WT	Control
A.2	WT	Control
B.1	WT	Drug
B.2	WT	Drug
D.1	KO	Drug
D.2	KO	Drug

Can’t use interaction term. We don’t know what the control effect would be in KO.

Exercise 2.1: Two-way ANOVA test

- Run differential analysis with 2-factor model

Practical considerations in downstream analysis

- *You will need to think carefully about what you want to learn from the data*
  - “I want to see all of the genes that changed” is not a sufficiently thought-out question
- What is my question ↔ which tests should I use
  - Anything that ever changes (union of different tests)
  - Effect of one factor depends on another (interaction term)
  - Consistent difference in Mutant treated vs Mutant untreated and WT treated (run a contrast: Mut treated vs (Mut untreated + WT treated)/2)

Practical considerations in downstream analysis

- Effects of different factors may be very different
  - May need to run separate analyses, splitting data based on the dominant factor
  - Example: tissues vs treatment – tissues usually much bigger effect
- Number of genes with interaction term effect may be much smaller than genes with effect due to factors independently

If you want to run contrasts

```
> metadata
       Genotype Gating
WT_A.1       WT      A
WT_A.2       WT      A
WT_B.1       WT      B
WT_B.2       WT      B
WT_C.1       WT      C
WT_C.2       WT      C
KO_A.1       KO      A
KO_A.2       KO      A
KO_B.1       KO      B
KO_B.2       KO      B
KO_C.1       KO      C
KO_C.2       KO      C
```

- It’s probably easier to set up contrasts in a single-factor design
- In R, you can paste the factors together to turn this into a one-factor model
- Then re-generate the model matrix (without intercept) and proceed with other steps

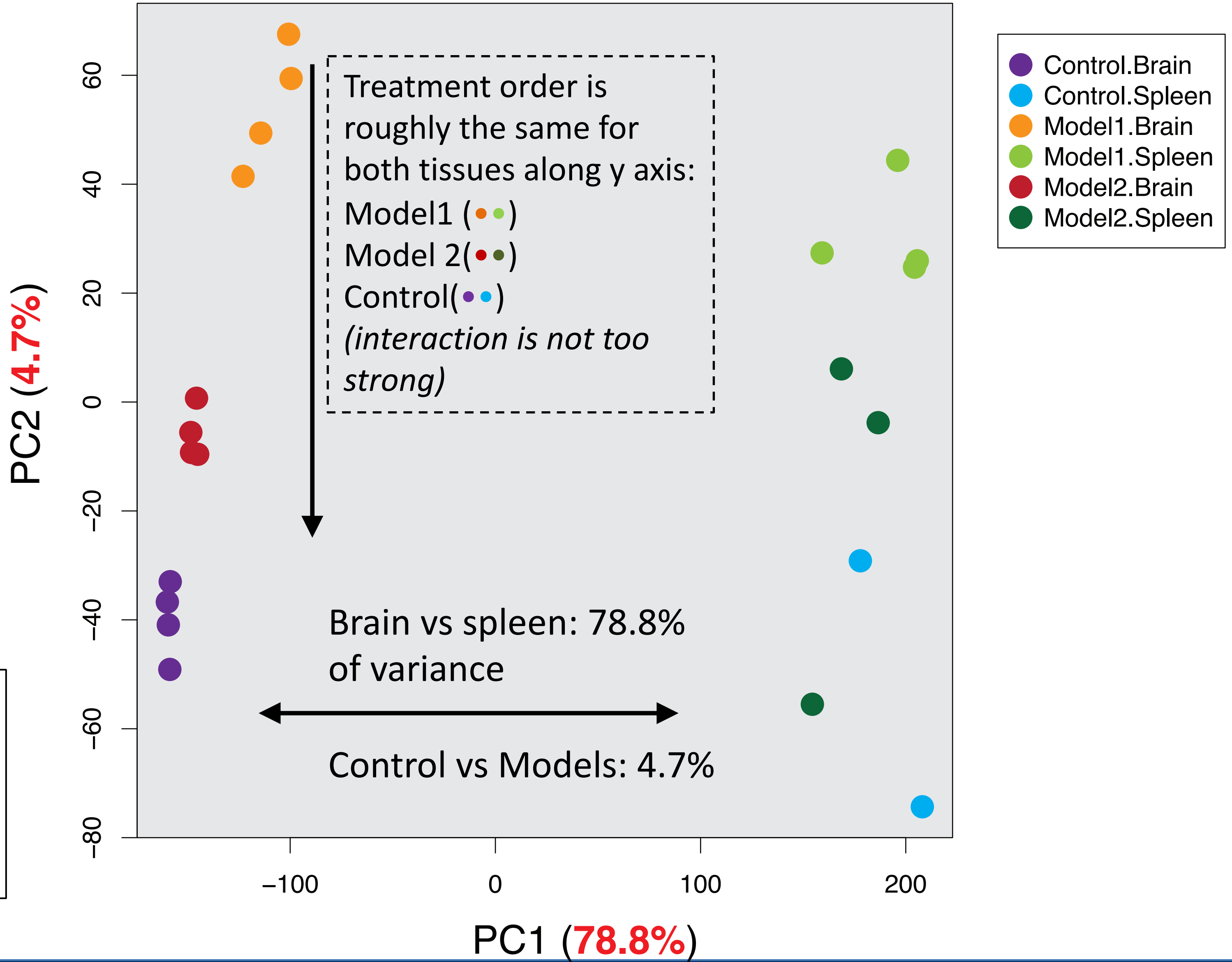
```
> metadata.oneFactor <- factor(paste(metadata[,1],metadata[,2],sep="."))

> metadata.oneFactor
 [1] WT.A WT.A WT.B WT.B WT.C WT.C KO.A KO.A KO.B KO.B KO.C KO.C
Levels: KO.A KO.B KO.C WT.A WT.B WT.C
```

PCA plot with multiple factors

- Factors often correlate with PC axes
- Assess the relative effect of each factor
- Visually inspect for interaction effects

At FDR<0.05:

- 11059 have effect for tissue
- 4883 genes have effect for treatment
- 2966 have effect for interaction term



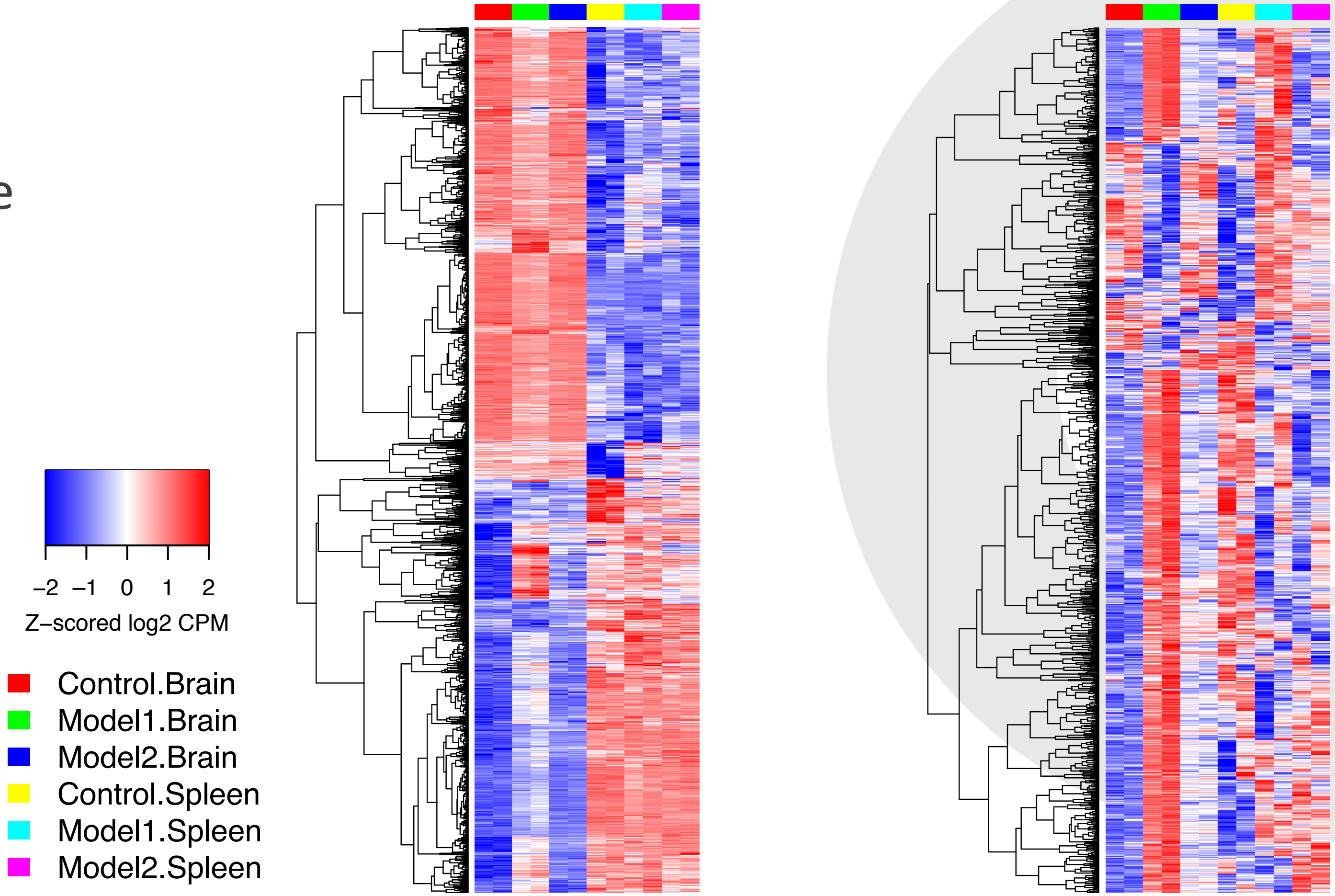
Treatment order is roughly the same for both tissues along y axis:
Model1 (● ●)
Model 2(● ●)
Control(● ●)
(interaction is not too strong)

Brain vs spleen: 78.8% of variance

Control vs Models: 4.7%

Heatmap with multiple factors

- Differences in the scale of effect sizes makes it hard to choose the right color scheme
  - Z-score dominant factors separately, plot in side-by-side heatmaps
- Significant interaction terms may be the most interesting genes, but may also be a minority of all DEGs
  - Plot separate heatmap for genes w/ significant interaction term



Normalized expression, log-scale and z-score as usual

Z-score each tissue separately, only show genes with significant interaction term

Exercise 2.2: Filtering strategies and heatmaps with two-way ANOVA

- Try different filtering strategies with two-way ANOVA stats

Repeated measures designs

- Repeated measurements on the same individual, need to control for individual-to-individual differences
- This is really a special case of a multi-way ANOVA: the “individual” is just another factor
- But: we’re looking for concordant effects across individuals
  - **Omit the interaction term**

Metadata table: repeated measures

Sample ID	Timepoint	Individual
A.1	1	A
A.2	2	A
A.3	3	A
A.4	4	A
B.1	1	B
B.2	2	B
B.3	3	B
B.4	4	B
...		

Exercise 2.3: Repeated measures

- Repeated measures differential

Other applications of advanced models in edgeR

- Alternative splicing: model changes in exon expression *relative* to gene expression
- DNA methylation by bisulfite sequencing: model changes in % methylation relative to overall CpG depth
- Metatranscriptomics (+ metagenomics): model changes in gene expression (from metatranscriptome) relative to gene abundance (from metagenome)

These models can get pretty complex, but they are good to get comfortable with.

Alternative splicing (exon usage)

- Quantify expression per *exon*, instead of per gene (see featureCounts manual for options)
- EdgeR version – see section 4.5 of user's manual
  - Include exon-to-gene annotation in DGEList with "genes" parameter
  - Use `diffSpliceDGE()` function: compares logFC for the exon to the logFC for the gene
- DESeq2/DEXSeq version
  - Include extra factor for exon vs gene-level expression, test for differences based on interaction term with biological factor
- Note: there are many methods for alternative splicing, and they often give different answers
  - rMATS: directly compare two alternative splicing events (e.g., skipped vs retained exon). Works from BAM files.

DNA methylation

- Quantify methylated and unmethylated counts per CpG
  - Results in 2 counts per sample (N samples = 2N columns in counts table)
- EdgeR – see section 4.7 of user's manual
  - Prepare metadata linking experimental group, sample, and methylation type
  - Make model matrix just of experimental group (e.g., Genotype)
  - Expand model matrix with `modelMatrixMeth()` to include sample and methylation type
  - Use contrasts to test for differences in *unmethylated* counts between experimental groups

Metadata table

Column	Genotype	Sample	Methyl
WT.1-M	WT	WT.1	Meth
WT.1-U	WT	WT.1	Unmeth
WT.2-M	WT	WT.2	Meth
WT.2-U	WT	WT.2	Unmeth
KO.1-M	KO	KO.1	Meth
KO.1-U	KO	KO.1	Unmeth
KO.2-M	KO	KO.2	Meth
KO.2-U	KO	KO.2	Unmeth

Metatranscriptomics

- We want to test if a gene is differentially expressed in a metatranscriptomics data set
  - Changes in taxonomic abundance will also affect measured changes in gene expression
  - Need to control for taxonomic abundance, which we could get from a metagenomics data set
- Quantify expression of each gene in meta-RNA-seq and meta-DNA-seq libraries from the same sample
- Prepare counts table of gene vs RNA-seq count and DNA-seq count for all sample
- Model changes in RNA-seq expression relative to DNA-seq

BREAK

Please complete our workshop survey

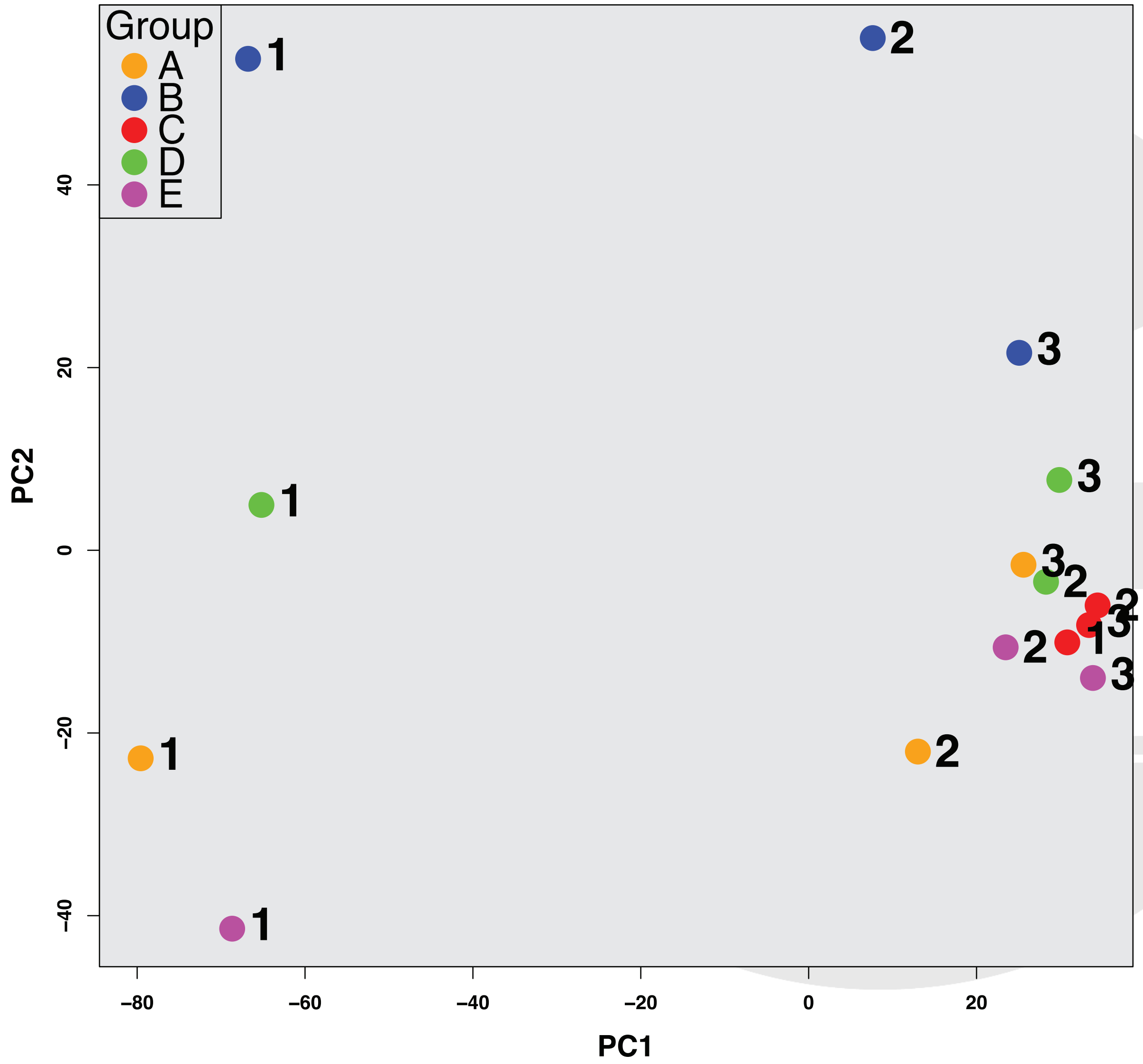
`http://go.uic.edu/RICWorkshopSurvey`

Additional topics

- Batch effects
- Removing outliers
- Testing for continuous variables
- What to do if you have no replicates *(besides getting more replicates)*
- Application to other –omics areas

Batch effects

- Factors that can affect quantitative measurements include:
  - Sequencing/array platform
  - Order of sample preparation
  - Lab/personnel preparing libraries
  - Protocol deviations: reagent batches, amount of material used, etc.
  - Instrument calibration
  - Animal housing conditions
- Assessing batch effect is easiest with PCA



Correcting for batch effect

- In differential analysis:
  - Set up batch as an additional factor
  - Include in the model, but don’t include any interaction terms
- In normalized expression:
  - Use removeBatchEffect() function from edgeR
  - Highly recommended to use log-scaled CPM
- To confirm:
  - Rerun PCA on batch-corrected normalized expression
  - Check BCV with and without batch factor
- **Must ensure that batch design is “complete”: some samples from each group in each batch is best**
  - **If batch is confounded with experimental group, then your experiment is unusable**

Original metadata

Sample ID	Treatment
A.1	A
A.2	A
B.1	B
B.2	B
C.1	C
C.2	C

Metadata with batch factor

Sample ID	Treatment	Batch
A.1	A	1
A.2	A	2
B.1	B	1
B.2	B	2
C.1	C	1
C.2	C	2

Exercise 2.4: Batch effect correction

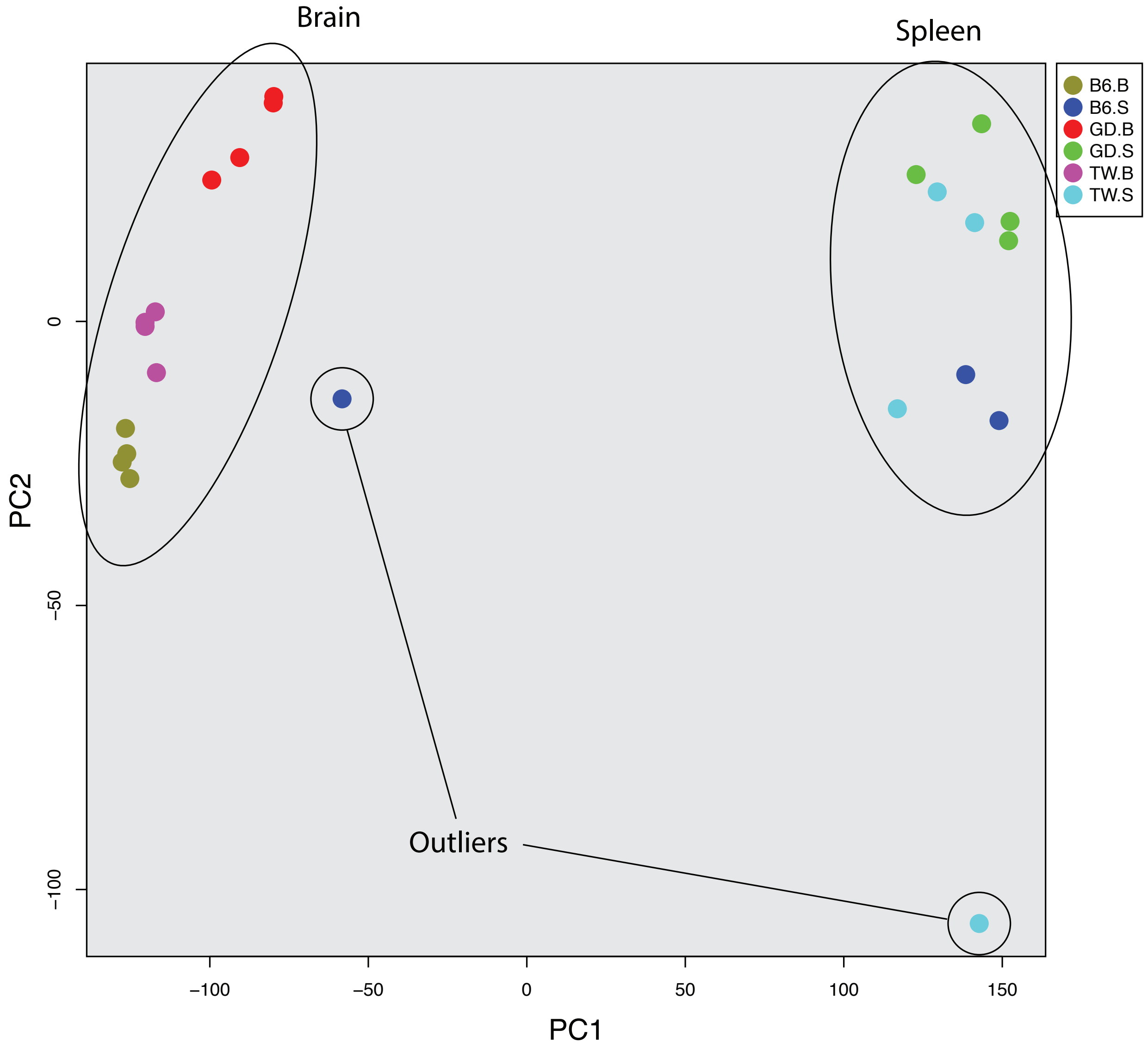
- Check and remove batch effect

Checking for outliers

- Assess similarly to batch effect: PCA
- Easiest way to remove an outlier: delete it from the metadata table
  - Rerun this step:

```
data <- data[,rownames(metadata)]
```

  - That sample will be dropped, and you can continue processing as usual
- Check dispersion/BCV before and after removing the outlier
  - If there’s no change (BCV changes less than ~5-10%), then don’t remove it
  - Keep in mind PCA is *relative similarity*



Using PCA: key points

- PCA is a very useful analysis for a **quick high-level experiment summary**
  - Check that samples are grouping together, and groups are separated
  - Effect size of different factors (qualitative)
  - Batch effects
  - Outlier samples
- **PCA does not tell you about absolute variability**, it is showing *relative variability*. Also check the dispersions/BCV.
- There may be better plots to include in a manuscript/poster, but always start with PCA as a diagnostic

Continuous variables

- Sometimes we want to model continuous covariates
  - Age
  - BMI
  - Clinical measurements
- We can also include continuous variables in edgeR
  - So far, we’ve only looked at categorical variables
- First you should examine the range of the variable
  - Are there a few extreme points that might disproportionately affect fits?
  - How much variation is there from sample-to-sample?
  - Would it make sense to re-scale the values (log-scale, etc.)?
  - Would it make sense to bin the values and treat as categorical? (e.g., high/medium/low)

Notes on correlating variables

- If you have complex metadata (i.e., from a clinical study), it's good to do exploratory analysis on metadata first
  - Looking for associations between your factors/variables
  - Consider binning continuous values, based on interpretation
- Also relevant between categorical variables
  - Example: Are most of the male patients black, and most of the female patients white?
- Tests/plots to run:
  - Continuous vs categorical: Kruskal-Wallis/Wilcox, boxplots
  - Categorical vs categorical: Fisher's Exact Test or Chi-Square
  - Continuous vs continuous: Pearson or Spearman correlation, scatterplots

Testing with continuous variables

Metadata

```
PatientID  Score  Disease
S.50       82     FALSE
S.51       90     FALSE
S.52       65     TRUE
S.53       90     TRUE
S.54       49     TRUE
S.55       43     TRUE
S.56       89     FALSE
S.57       44     TRUE
S.58       85     FALSE
```

```
score <- as.numeric(metadata[,1])
disease <- factor(metadata[,2])
```

```
> model.matrix(~score+disease)
  (Intercept) score diseaseTRUE
1           1    82           0
2           1    90           0
3           1    65           1
4           1    90           1
5           1    49           1
6           1    43           1
7           1    89           0
8           1    44           1
9           1    85           0
```

- Include continuous variables as numeric vectors in your model matrix
  - There will always be 1 coefficient for the continuous variable
  - Actual values instead of dummy variables
  - With or without interaction term
- Test with glmQLFTest on that coefficient index (e.g., coef=2 below)

Score p-value:

```
qlf <- glmQLFTest(fit, coef=2)
```

Disease p-value:

```
qlf <- glmQLFTest(fit, coef=3)
```

Testing with continuous variables

Metadata

```
PatientID  Score  Disease
S.50       82     FALSE
S.51       90     FALSE
S.52       65     TRUE
S.53       90     TRUE
S.54       49     TRUE
S.55       43     TRUE
S.56       89     FALSE
S.57       44     TRUE
S.58       85     FALSE
```

```
score <- as.numeric(metadata[,1])
disease <- factor(metadata[,2])
```

```
> model.matrix(~score+disease+score:disease)
  (Intercept) score diseaseTRUE score:diseaseTRUE
1           1    82           0                 0
2           1    90           0                 0
3           1    65           1                65
4           1    90           1                90
5           1    49           1                49
6           1    43           1                43
7           1    89           0                 0
8           1    44           1                44
9           1    85           0                 0
```

- Include continuous variables as numeric vectors in your model matrix
  - There will always be 1 coefficient for the continuous variable
  - Actual values instead of dummy variables
  - With or without interaction term
- Test with glmQLFTest on that coefficient index (e.g., coef=2 below)

Score p-value:

```
qlf <- glmQLFTest(fit, coef=2)
```

Disease p-value:

```
qlf <- glmQLFTest(fit, coef=3)
```

Interaction p-value:

```
qlf <- glmQLFTest(fit, coef=4)
```

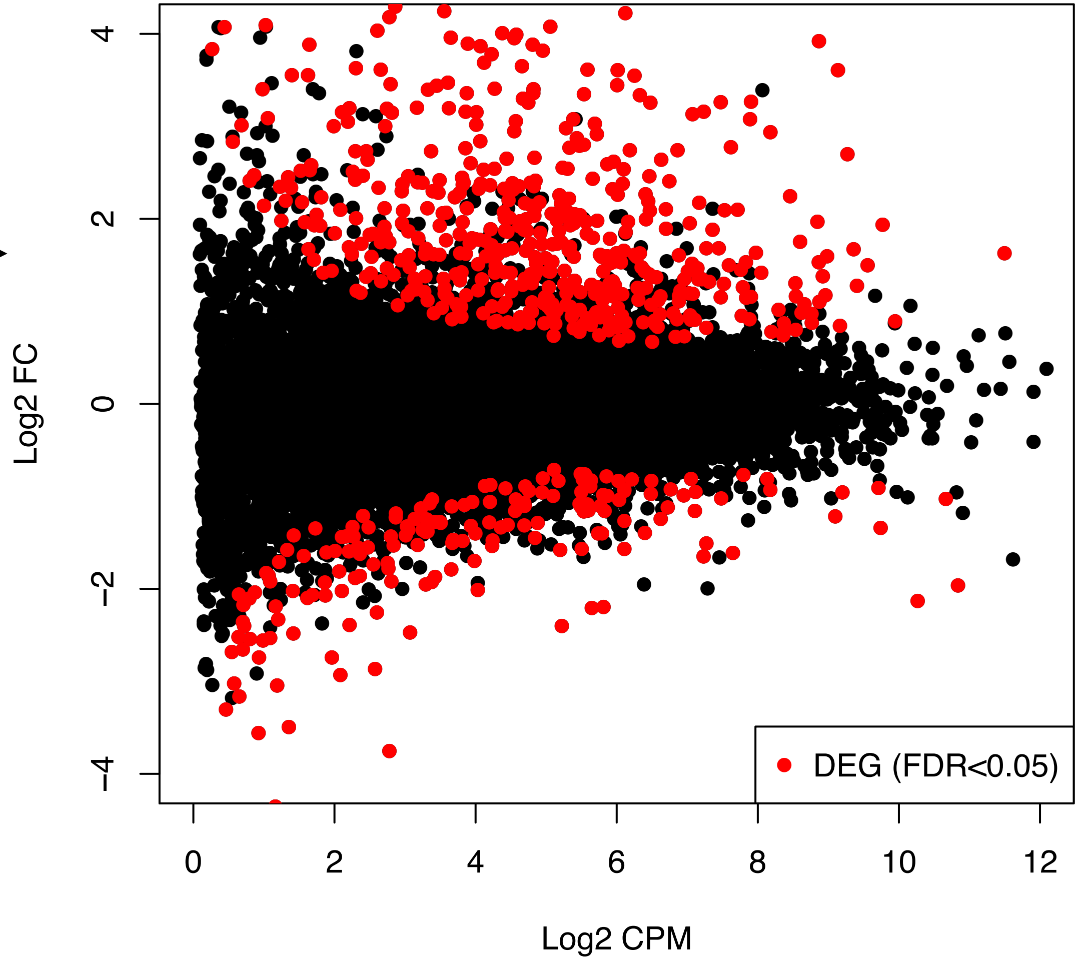
Exercise 2.5: Test with a continuous variable

- Try differential expression with a continuous variable

What to do if you have no replicates

- **This is not advised**
- But, sometimes samples are rare and you just need a more qualitative analysis
- **Don’t just use fold-change:** this biases you towards low-expressed genes →
- Best approach:
  - Process samples as normal, but instead of estimating dispersion, set it to a fixed value
  - Choose something reasonable from your experience
  - Use q-value to *prioritize* genes, keep in mind that the “significance” is artificial

Low expressed genes generally have bigger fold-changes, but are not as statistically significant



Exercise 2.6: Differential with no replicates

- Try a differential expression with a fixed dispersion
- ***NOTE: bonus exercises 2.7 (Gene ID conversion with biomaRt)***

Applications to other –omics areas

- Isoform expression
- ATAC-seq peaks
- ChIP-seq peaks
- Bisulfite-seq
- Metagenomics

Applications to other –omics areas

- **Isoform expression**
- ATAC-seq peaks
- ChIP-seq peaks
- Bisulfite-seq
- Metagenomics

- Quantification (e.g., Kallisto or Salmon) will yield *estimated* counts
  - E.g., 143.56 instead of 143
- But estimated counts are still representative of the number of copies of the transcript in the library
- **Use as-is in edgeR; edgeR will round the counts to integers**
  - You may want to compare isoform to gene logFCs, as noted earlier for alternative splicing of exons
- *Related analysis noted earlier: alternative exon usage*

Applications to other –omics areas

- Isoform expression
- **ATAC-seq peaks**
- ChIP-seq peaks
- Bisulfite-seq
- Metagenomics

- Map to genome (BWA MEM)
- Call peaks (Macs2)
- Combine peaks across samples (bedtools)
- Quantify BAM files to get counts table (featureCounts)
- **The counts table can be used directly in edgeR** (but, you might have >100,000 peaks)

Applications to other –omics areas

- Isoform expression
- ATAC-seq peaks
- **ChIP-seq peaks**
- Bisulfite-seq
- Metagenomics

- Similar to ATAC-seq, but we also have the input sample:
  - Peak calling, combine peaks
  - Quantify BAM files to get counts table – *for both ChIP and input samples*
- Option A:
  - Subtract input counts from ChIP counts, adjusting for library size
  - **The input-subtracted counts table can be used in edgeR**
- Option B:
  - **Model ChIP vs input counts in interaction term with experimental variable in edgeR**

Applications to other –omics areas

- Isoform expression
- ATAC-seq peaks
- ChIP-seq peaks
- **Bisulfite-seq**
- Metagenomics

- Quantification will give you methylated and unmethylated counts per CpG
- Option A:
  - Model changes in counts relative to overall depth, as noted earlier
- Option B:
  - Take log-ratio of methylated/unmethylated (“M-value”), run test in Limma
    - Limma set-up is similar to edgeR, but uses normal model
    - Important to remove CpGs with low coverage first

Applications to other –omics areas

- Isoform expression
- ATAC-seq peaks
- ChIP-seq peaks
- Bisulfite-seq
- **Metagenomics**

- For both 16S and shotgun
- Quantification will yield counts per taxon, or per gene/pathway
- May need to do some filtering:
  - Split counts tables by kingdom – abundances and variability may be very different
  - Keep in unassigned taxa for normalization?
  - Remove mitochondrial/chloroplasts from 16S
- Consider normalization strategies, effect of TMM
- For metatranscriptomics, will also need metagenomics data to distinguish gene expression vs abundance changes, as noted earlier
- **Otherwise, use as-is in edgeR**

Reminders for metadata table

- Best to store metadata in a separate table
  - Formatting sample names with metadata becomes awkward with a lot of factors
  - Do not rely on colors or special formatting in Excel to differentiate groups. Add a factor/column instead.
    - These are lost when exported to tabular, or imported in R, and you can’t sort or filter on them in Excel
- Pick simple sample names
  - Stick to alphanumeric (A-Z, 0-9), and underscore (_) and period (.)
  - **Avoid dash (-):** R turns these into periods for column names
  - Avoid spaces and other special characters (+/*&?%$@!<>;:’”)
- Reference guide:

https://github.com/uic-ric/uic-ric.github.io/wiki/Providing-sample-information

Summary

- EdgeR is a powerful and flexible differential statistics tool
  - Pair-wise comparisons
  - Multi-group and multi-factor comparisons
  - Batch effects
  - Applicable to any -omics area where features are quantified by *counting reads*
- Complement with PCA and heatmaps to get a visual understanding of what's going on
  - Also look at the dispersion/BCV
- Similar R package that uses normal models: limma
  - Same authors as edgeR, similar usage
  - Used for microarray, normalized & quantitative metabolomics and proteomics
  - Used by edgeR for variance squeezing
- More reading: edgeR user's guide

https://www.bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf

THANK YOU!

Please complete our workshop survey TODAY:

http://go.uic.edu/RICWorkshopSurvey