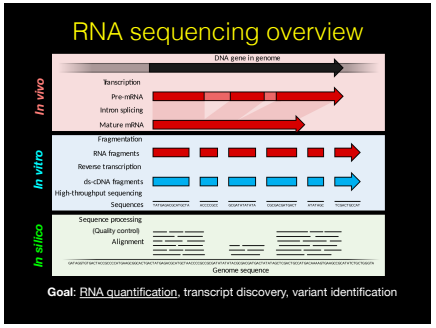
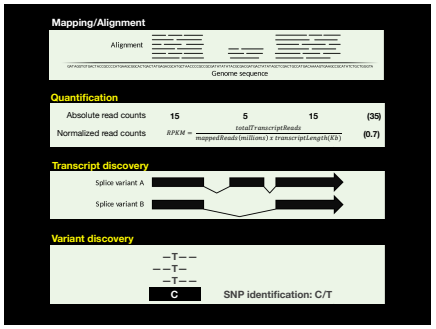
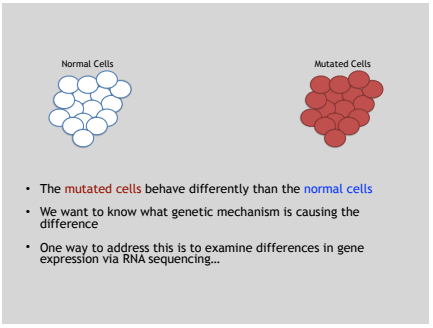
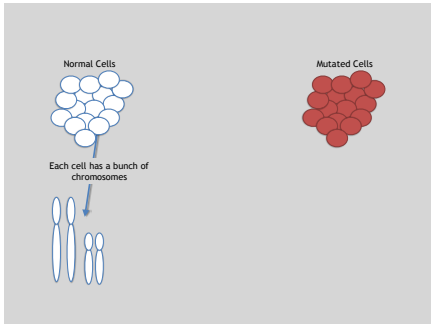


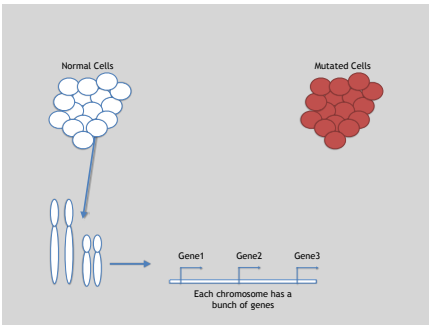


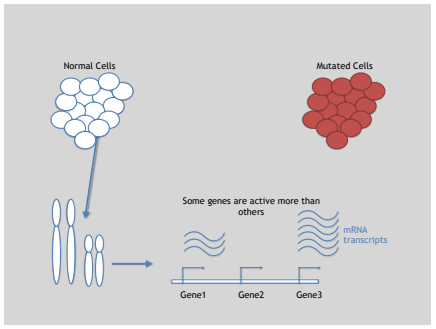
RNA Sequencing

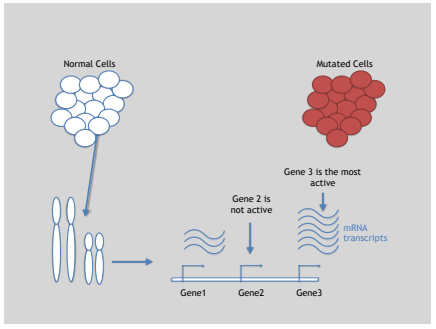
The absolute basics

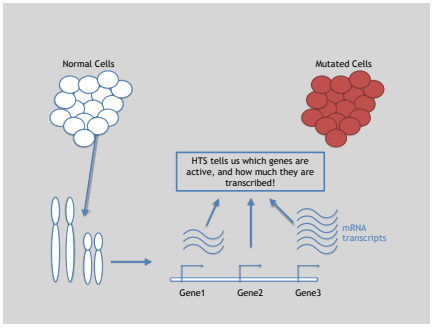


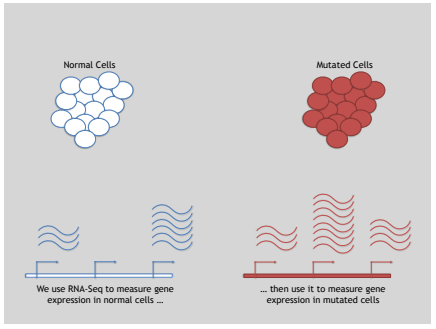


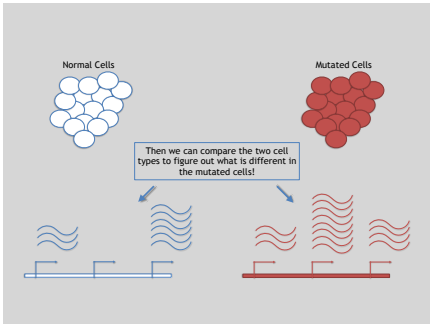


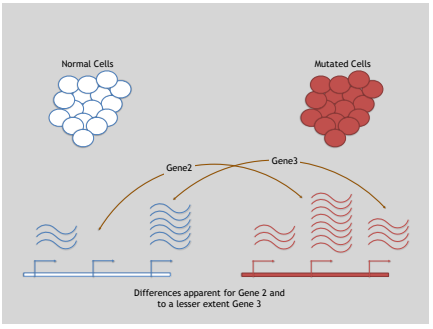












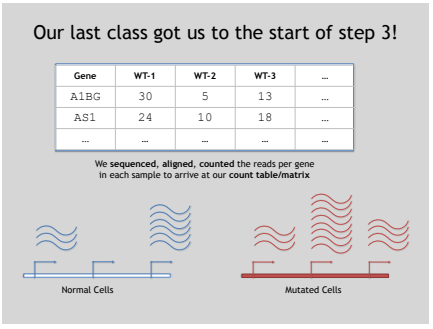
3 Main Steps for RNA-Seq:

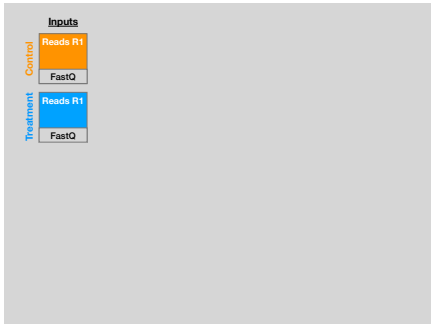
1) **Prepare a sequencing library**
   (RNA to cDNA conversion via reverse transcription)

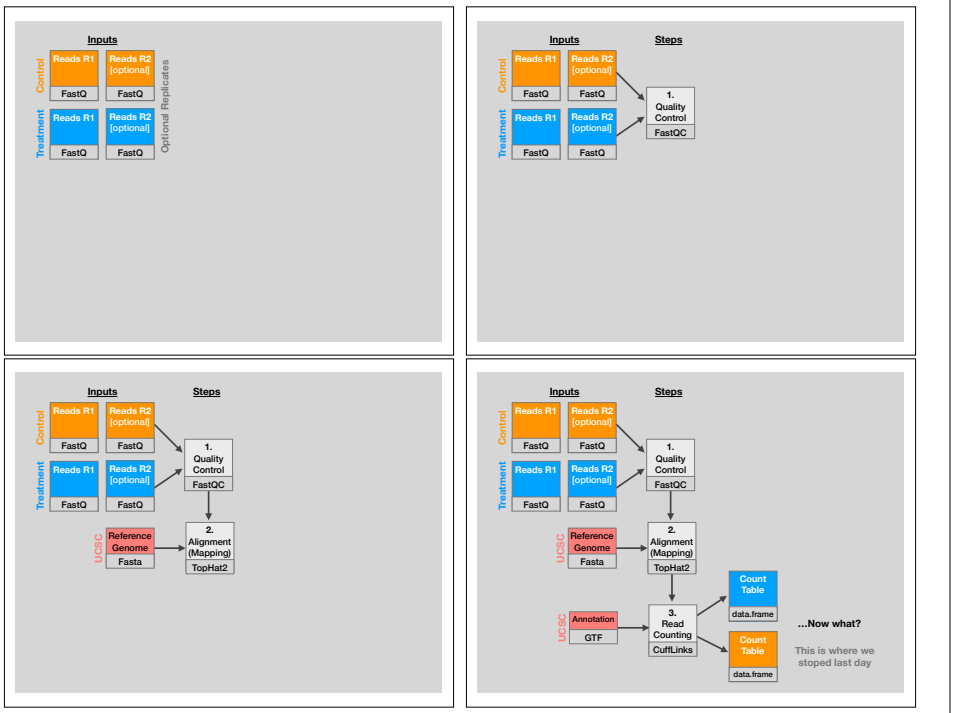
2) **Sequence**
   (Using the same technologies as DNA sequencing)

3) **Data analysis**
   (Often the major bottleneck to overall success!)

We will discuss each of these steps - but we will focus on step 3 today!



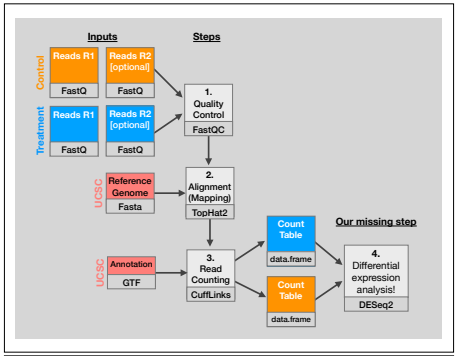



Inputs
Control
Reads R1
Reads R2 [optional]
FastQ
Treatment
Optional Replicates

Inputs
Steps
Control
Reads R1
Reads R2 [optional]
FastQ
Treatment
1. Quality Control
FastQC

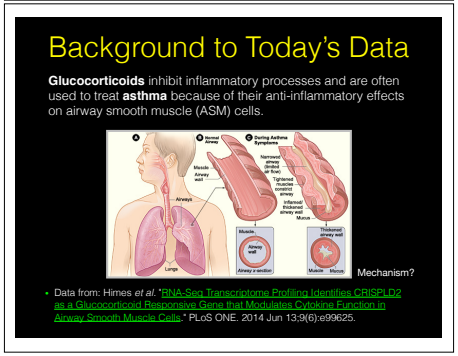
Inputs
Steps
Control
Reads R1
Reads R2 [optional]
FastQ
Treatment
1. Quality Control
FastQC
UCSC
Reference Genome
Fasta
2. Alignment (Mapping)
TopHat2

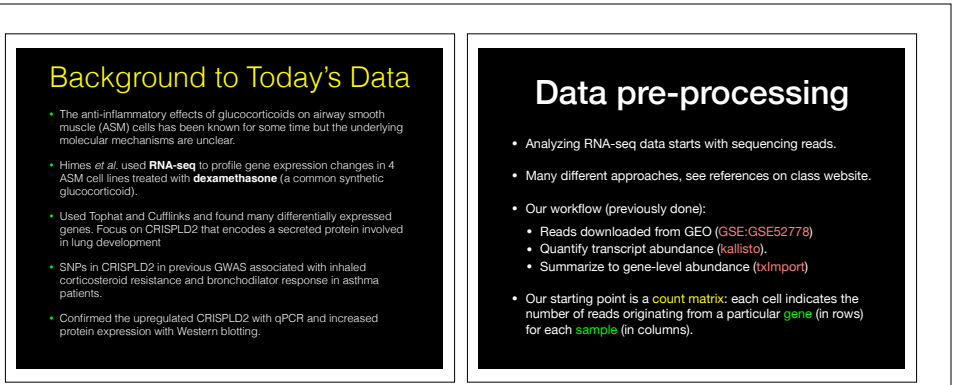
Inputs
Steps
Control
Reads R1
Reads R2 [optional]
FastQ
Treatment
1. Quality Control
FastQC
UCSC
Reference Genome
Fasta
2. Alignment (Mapping)
TopHat2
Annotation
GTF
3. Read Counting
CuffLinks
Count Table
data.frame
...Now what?
This is where we stoped last day


Inputs
Steps
Control
Reads R1
Reads R2 [optional]
FastQ
Treatment
1. Quality Control
FastQC
UCSC
Reference Genome
Fasta
2. Alignment (Mapping)
TopHat2
Annotation
GTF
3. Read Counting
CuffLinks
Count Table
data.frame
Our missing step
4. Differential expression analysis!
DESeq2


Do it Yourself!
Install DESeq2
Bioconductor Setup Link
install.packages("BiocManager")
BiocManager::install()
For this class, you'll also need DESeq2:
BiocManager::install("DESeq2")
Note: Answer NO to prompts to install from source or update...

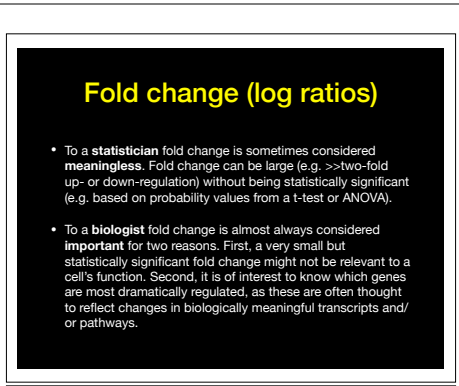

Do it Yourself!
Old packages: 'devtools', 'dplyr', 'DT', 'ggplot2', 'ggpubr', 'lattice', 'MASS', 'Matrix', 'mclust', 'mgcv', 'openssl', 'packrat', 'pkgload', 'ps', 'psych', 'raster', 'rcmdcheck', 'Rcpp', 'remotes', 'rsconnect', 'sessioninfo', 'shiny', 'shinythemes', 'survival', 'tidyr', 'tinytex', 'xfun'
Update all/some/none? [a/s/n]:
n
Install DESeq2
Bioconductor Setup Link
install.packages("BiocManager")
BiocManager::install()
For this class, you'll also need DESeq2:
BiocManager::install("DESeq2")
Note: Answer NO to prompts to install from source or update...

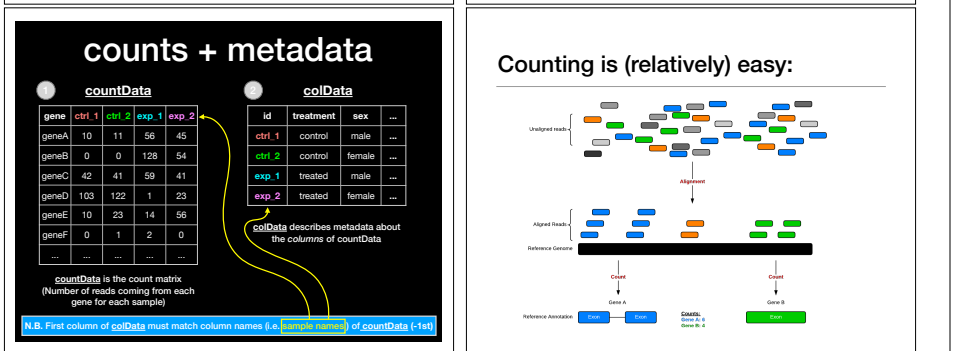

Background to Today's Data
Glucocorticoids inhibit inflammatory processes and are often used to treat asthma because of their anti-inflammatory effects on airway smooth muscle (ASM) cells.
Mechanism?
Data from: Himes et al. "RNA-Seq Transcriptome Profiling Identifies CRISPLD2 as a Glucocorticoid Responsive Gene that Modulates Cytokine Function in Airway Smooth Muscle Cells." PLoS ONE. 2014 Jun 13;9(6):e99625.


Background to Today's Data
The anti-inflammatory effects of glucocorticoids on airway smooth muscle (ASM) cells has been known for some time but the underlying molecular mechanisms are unclear.
Himes et al. used RNA-seq to profile gene expression changes in 4 ASM cell lines treated with dexamethasone (a common synthetic glucocorticoid).
Used Tophat and Cufflinks and found many differentially expressed genes. Focus on CRISPLD2 that encodes a secreted protein involved in lung development
SNPs in CRISPLD2 in previous GWAS associated with inhaled corticosteroid resistance and bronchodilator response in asthma patients.
Confirmed the upregulated CRISPLD2 with qPCR and increased protein expression with Western blotting.

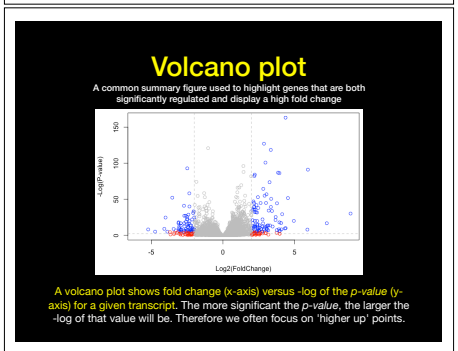
Data pre-processing
Analyzing RNA-seq data starts with sequencing reads.
Many different approaches, see references on class website.
Our workflow (previously done):
Reads downloaded from GEO (GSE:GSE52778)
Quantify transcript abundance (kallisto).
Summarize to gene-level abundance (txImport)
Our starting point is a count matrix: each cell indicates the number of reads originating from a particular gene (in rows) for each sample (in columns).

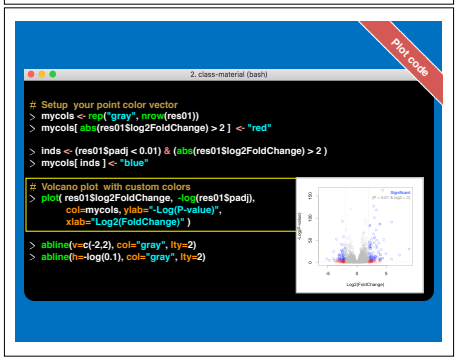

Do it Yourself!
Hands-on time!
https://bioboot.github.io/bimm143_F19/lectures/#14

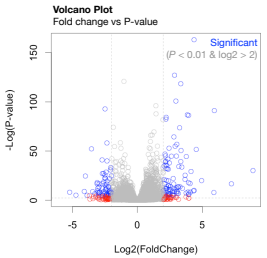

Fold change (log ratios)
To a statistician fold change is sometimes considered meaningless. Fold change can be large (e.g. >>two-fold up- or down-regulation) without being statistically significant (e.g. based on probability values from a t-test or ANOVA).
To a biologist fold change is almost always considered important for two reasons. First, a very small but statistically significant fold change might not be relevant to a cell's function. Second, it is of interest to know which genes are most dramatically regulated, as these are often thought to reflect changes in biologically meaningful transcripts and/or pathways.


counts + metadata
countData
gene | ctrl_1 | ctrl_2 | exp_1 | exp_2
geneA | 10 | 11 | 56 | 45
geneB | 0 | 0 | 128 | 54
geneC | 42 | 41 | 59 | 41
geneD | 103 | 122 | 1 | 23
geneE | 10 | 23 | 14 | 56
geneF | 0 | 1 | 2 | 0
countData is the count matrix (Number of reads coming from each gene for each sample)
colData
id | treatment | sex | ...
ctrl_1 | control | male | ...
ctrl_2 | control | female | ...
exp_1 | treated | male | ...
exp_2 | treated | female | ...
colData describes metadata about the columns of countData
N.B. First column of colData must match column names (i.e. sample names) of countData (-1st)

Counting is (relatively) easy:
Unaligned reads
Alignment
Aligned Reads
Reference Genome
Count
Gene A
Gene B
Reference Annotation
Exon


Volcano plot
A common summary figure used to highlight genes that are both significantly regulated and display a high fold change
-Log(P-value)
Log2(FoldChange)
A volcano plot shows fold change (x-axis) versus -log of the p-value (y-axis) for a given transcript. The more significant the p-value, the larger the -log of that value will be. Therefore we often focus on 'higher up' points.


Plot code
2. class-material (bash)
Setup your point color vector
> mycols <- rep("gray", nrow(res01))
> mycols[abs(res01$log2FoldChange) > 2] <- "red"
> inds <- (res01$padj < 0.01) & (abs(res01$log2FoldChange) > 2)
> mycols[inds] <- "blue"
Volcano plot with custom colors
> plot(res01$log2FoldChange, -log(res01$padj),
col=mycols, ylab="-Log(P-value)",
xlab="Log2(FoldChange)")
> abline(v=c(-2,2), col="gray", lty=2)
> abline(h=-log(0.1), col="gray", lty=2)



Recent developments in RNA-Seq

- **Long read sequences:**
  - PacBio and Oxford Nanopore [Recent Paper]
- **Single-cell RNA-Seq**: [Review article]
  - Observe heterogeneity of cell populations
  - Detect sub-population
- **Alignment-free quantification**:
  - Kallisto [Software link]
  - Salmon [Software link, Blog post]

Reference

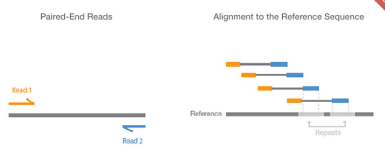
Additional Reference Slides

Reference

Public RNA-Seq data sources

- **Gene Expression Omnibus** (GEO):
  - http://www.ncbi.nlm.nih.gov/geo/
  - Both microarray and sequencing data
- **Sequence Read Archive** (SRA):
  - http://www.ncbi.nlm.nih.gov/sra
  - All sequencing data (not necessarily RNA-Seq)
- **ArrayExpress**:
  - https://www.ebi.ac.uk/arrayexpress/
  - European version of GEO
- All of these have links between them

Reference



Paired-end sequencing enables both ends of the DNA fragment to be sequenced. Because the distance between each paired read is known, alignment algorithms can use this information to map the reads over repetitive regions more precisely. This results in much better alignment of the reads, especially across difficult-to-sequence, repetitive regions of the genome.

Taken From: https://www.illumina.com/science/technology/next-generation-sequencing/paired-end-vs-single-read-sequencing.html

Reference

Count Normalization

- Normalization is required to make comparisons in gene expression
  - Between 2+ genes in one sample
  - Between genes in 2+ samples
- Genes will have more reads mapped in a sample with high coverage than one with low coverage
  - 2x depth ≈ 2x expression
- Longer genes will have more reads mapped than shorter genes
  - 2x length ≈ 2x more reads

Reference

Normalization: RPKM, FPKM & TPM

- **N.B.** Some tools for differential expression analysis such as edgeR and **DESeq2** want raw read counts - i.e. non normalized input!
- However, often for your manuscripts and reports you will want to report normalized counts
- RPKM, FPKM and TPM all aim to normalize for sequencing depth and gene length. For the former:
  - Count up the total reads in a sample and divide that number by 1,000,000 - this is our "per million" scaling.
  - Divide the read counts by the "per million" scaling factor. This normalizes for sequencing depth, giving you reads per million (RPM)
  - Divide the RPM values by the length of the gene, in kilobases. This gives you RPKM.

Reference

- FPKM was made for paired-end RNA-seq
- With paired-end RNA-seq, two reads can correspond to a single fragment
- The only difference between RPKM and FPKM is that FPKM takes into account that two reads can map to one fragment (and so it doesn't count this fragment twice).

Reference

- **TPM** is very similar to RPKM and FPKM. The only difference is the order of operations:
  - First divide the read counts by the length of each gene in kilobases. This gives you reads per kilobase (RPK).
  - Count up all the RPK values in a sample and divide this number by 1,000,000. This is your "per million" scaling factor.
  - Divide the RPK values by the "per million" scaling factor. This gives you TPM.
- Note, the only difference is that you normalize for gene length first, and then normalize for sequencing depth second.

Reference

- When you use TPM, the sum of all TPMs in each sample are the same.
- This makes it easier to compare the proportion of reads that mapped to a gene in each sample.
- In contrast, with RPKM and FPKM, the sum of the normalized reads in each sample may be different, and this makes it harder to compare samples directly.