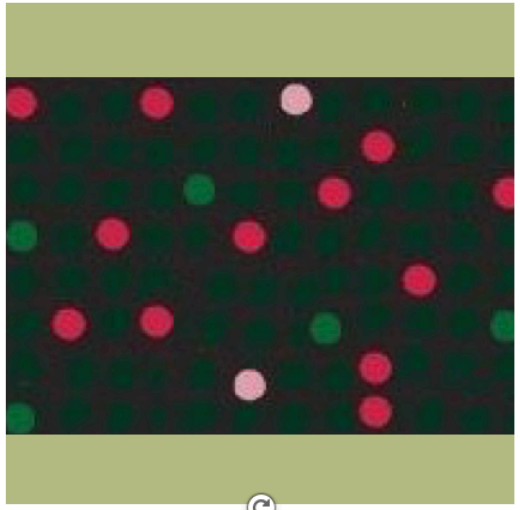
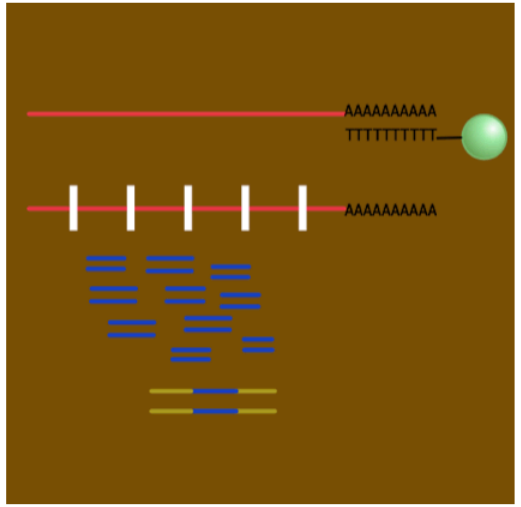
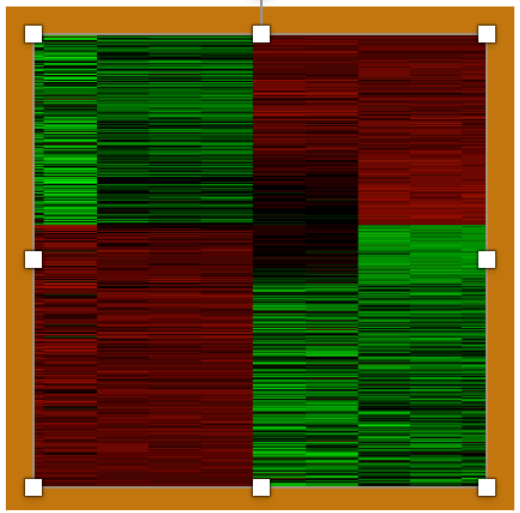
Whole Transcriptomic Analysis

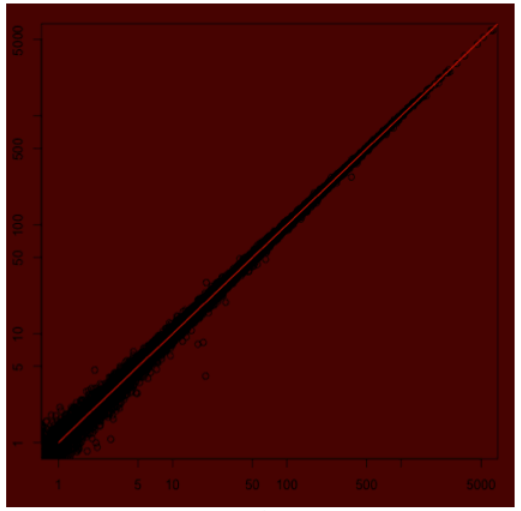
Manjula Thimma, AlaguRaj Veluchamy, Arun P Nagarajan, Robert Lehmann, Octavio

RNAseq for differential gene expression analysis





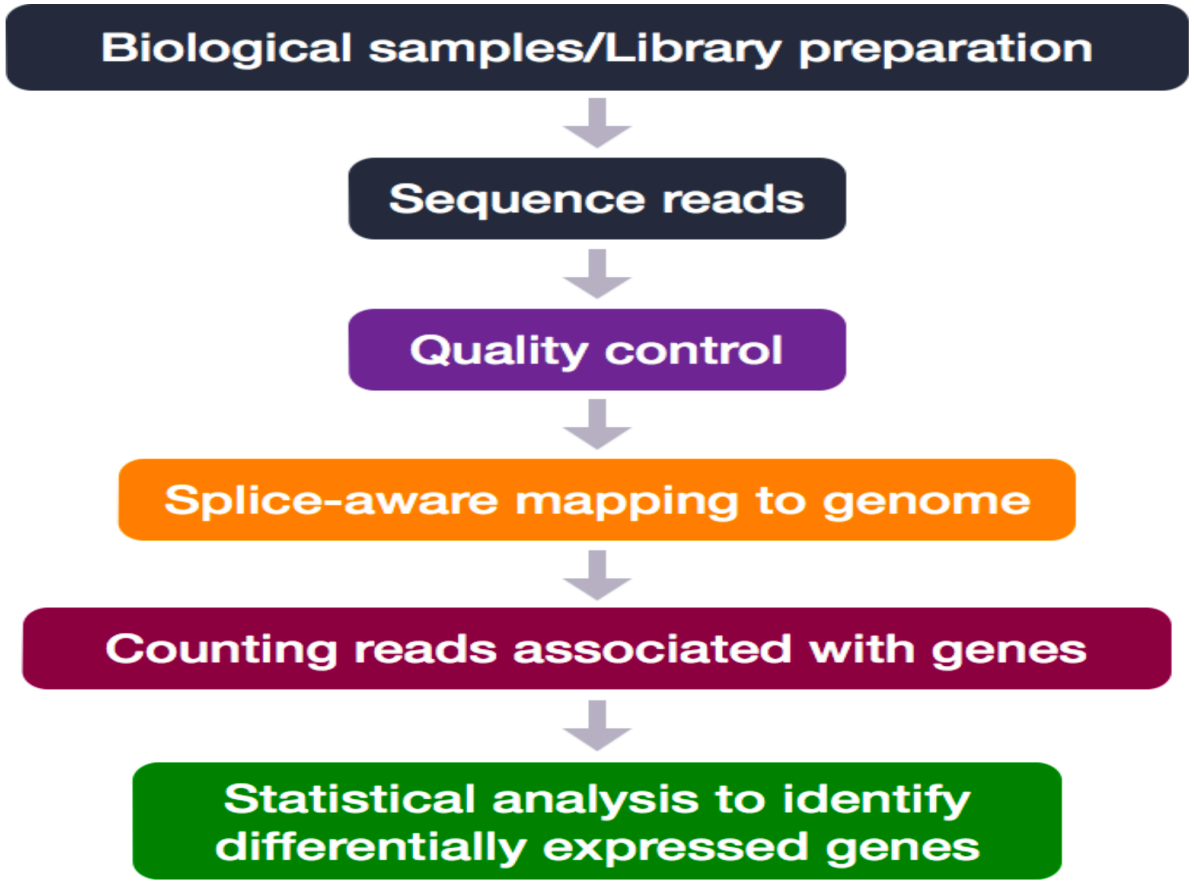




Manjula Thimma

Prof. Jesper Tegner

RNA-Seq Analysis Workflow



A note on P-values

The problem of multiple hypothesis testing

FDR = False Discovery Rate
Statisticians have been harping on something called P-value fishing for a long time.
If you keep running tests over and over, you'll eventually **get something** to come up as **significant by random chance**.

How do you fix this?

Adjust your P-values to q-values

Benjamini & Hochberg (BH):

rank genes by P-value (large to small)
each P-value is adjusted by the
number of genes with a
smaller P-value than itself

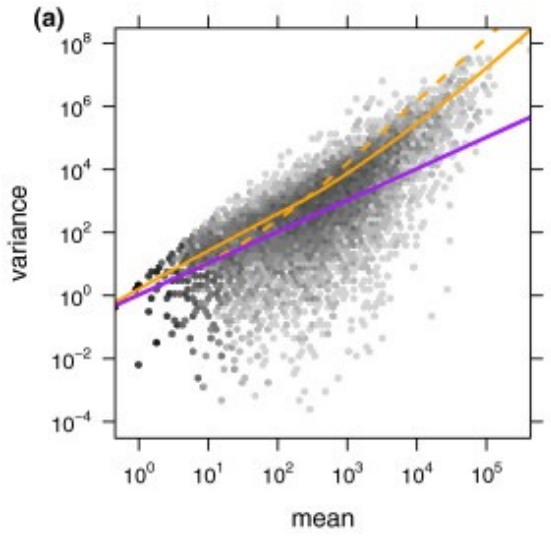
```
q-value = p-value * n/(n-k)

n = number of genes
k = rank in  genelist
```

Statistical Distributions

gaussian, poisson, **negative binomial** -- what does all this mean?

RNA-seq data fits a Negative Binomial (NB) distribution.
But really, that's just saying that RNAseq looks like "counts" data with more variation than just statistical fluctuations– it also has biological variation in it.

How do we know? Because, when you measure variance (per gene, between replicates), it's not equal to the mean, and it's not even a good linear fit



poisson
negative binomial fit

* For real data, even the NB fit isn't always great

Anders & Huber, Genome Biology 2010

So, what do I do now?

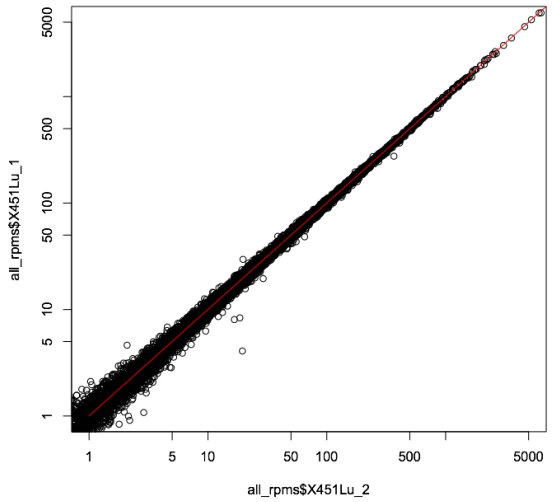
A little QC & a lot of downstream analysis

- QC
  - How well do the replicates correlate with each other?
  - Does a PCA plot show that my samples group by genotype?
  - What fraction of transcripts are expressed > 1 RPKM?

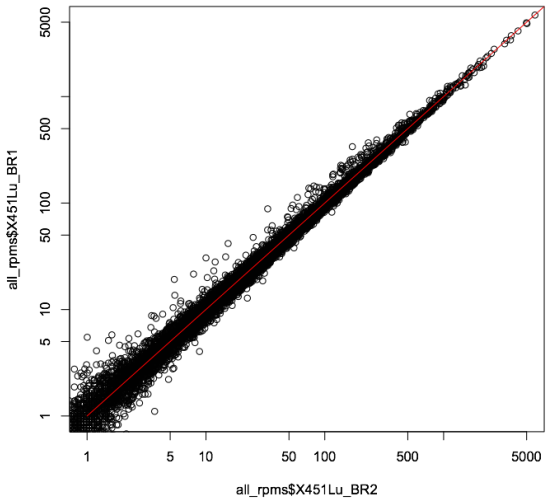
- Downstream analysis
  - make lots of Excel tables of comparisons
  - make some heatmaps (in R or using gplots, CummeRbund)
  - figure out if any pathways are enriched among genes that change
    (DAVID, GSEA, PathDE, etc)

RNAseq post-analysis QC
RNAseq Replicate & Sample Correlations

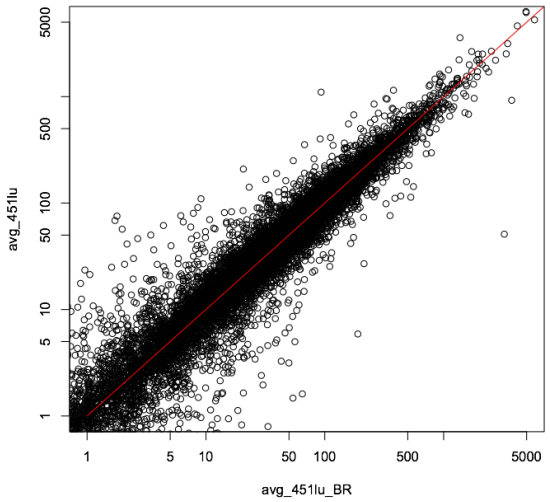
Replicates, Condition 1



Replicates, Condition 2



Condition 1 Vs. Condition 2



All axes show RPM
(reads per million mapped)

Input to your analysis pipeline

RNA-Seq Workflow: Count matrix

```
wt_rawcounts <- read.csv("fibrosis_wt_rawcounts.csv")
```

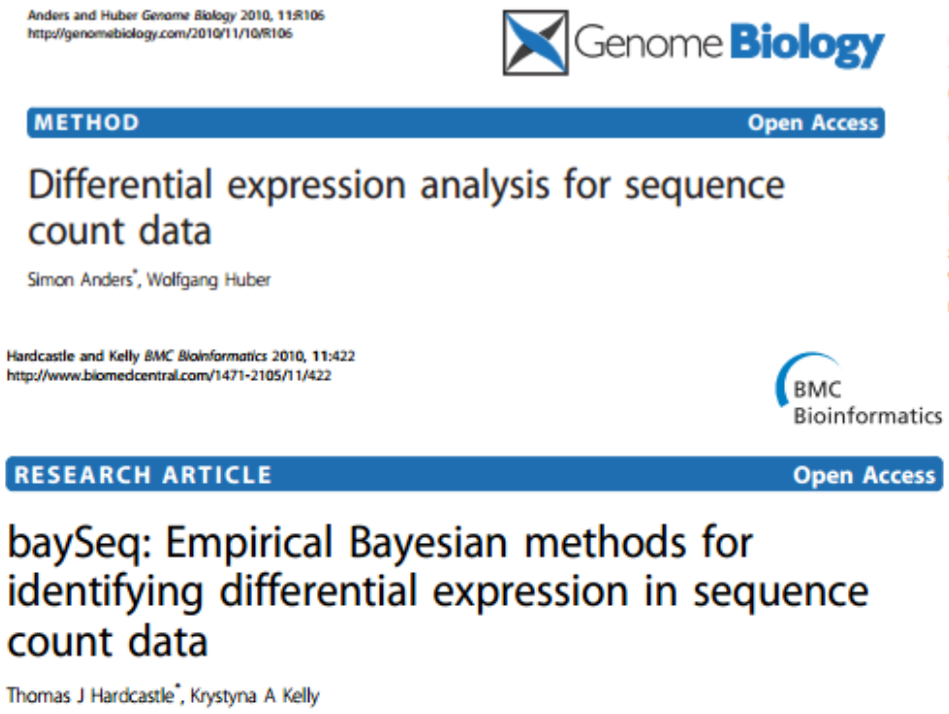
	wt_normal1	wt_normal2	wt_normal3	wt_fibrosis1	wt_fibrosis2	wt_fibrosis3	wt_fibrosis4
ENSMUSG00000102693	0	0	0	0	0	0	0
ENSMUSG00000064842	0	0	0	0	0	0	0
ENSMUSG00000051951	3	1	1	42	52	16	35
ENSMUSG00000102851	0	0	0	0	0	0	0
ENSMUSG00000103377	0	0	0	0	0	0	0
ENSMUSG00000104017	0	0	0	0	0	0	0
ENSMUSG00000103025	0	0	0	1	0	0	0
ENSMUSG00000089699	0	0	0	0	0	0	0
ENSMUSG00000103201	0	0	0	0	0	0	0
ENSMUSG00000103147	0	0	0	0	1	1	1

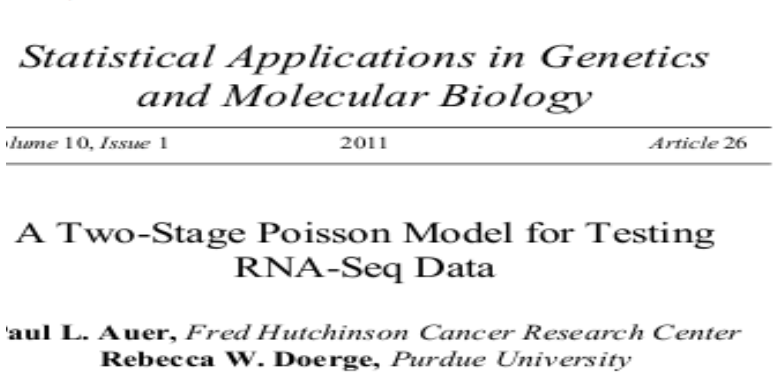
Output from statistical anlaysis

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
MOV10	21681.7998	4.7695983	0.10269615	46.232357	0.000000e+00	0.000000e+00
H1F0	7881.0811	1.5250811	0.05548216	27.479961	3.047848e-166	2.489330e-162
HIST1H1C	1741.3830	1.4868361	0.06844630	21.700664	2.022230e-104	1.101104e-100
TXNIP	5133.7486	1.3868320	0.06759178	20.513587	1.628305e-93	6.649590e-90
NEAT1	21973.7061	0.9087853	0.04601897	19.747620	8.408861e-87	2.747175e-83
KLF10	1694.2109	1.2093969	0.06339756	19.067600	4.693529e-81	1.277813e-77
INSIG1	11872.5106	1.2260848	0.06780306	18.079993	4.581384e-73	1.069099e-69
NR1D1	969.9119	1.5236259	0.08754050	17.359140	1.682239e-67	3.434921e-64
WDFY1	1422.7361	1.0629160	0.06251739	16.996459	8.723327e-65	1.583284e-61
HSPA1A	31481.9954	0.8800184	0.05216017	16.870952	7.360074e-64	1.202268e-60
HSPA6	168.2522	4.4993734	0.17982421	16.437244	1.035213e-60	1.537291e-57
HMGCS1	11833.0545	0.9107052	0.05653766	16.106656	2.290806e-58	3.118359e-55
HSPA1B	29876.3391	0.8164195	0.05203463	15.689470	1.785400e-55	2.243424e-52
LAMC1	5683.4671	0.9144938	0.05832194	15.681609	2.020714e-55	2.357740e-52
TMCO1	1718.7579	0.9358767	0.06016436	15.554555	1.481817e-54	1.613699e-51
ADAMTS1	9567.0703	1.0083996	0.06693542	15.063332	2.821975e-51	2.881060e-48

Count-based methods (R packages)

1. **DESeq** -- based on negative binomial distribution
2. **edgeR** -- use an overdispersed Poisson model
3. **baySeq** -- use an empirical Bayes approach
4. **TSPM** -- use a two-stage poisson model
5. **limmewrbloom** – bit less sensitive for smaller size samples





Vignette for help with package

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Home | Install | Help

Home » Bioconductor 3.6 » Software Packages » DESeq2

DESeq2

platforms all | downloads top 5% | posts 353 / 1 / 3 / 60 | in Bioc 5 years
build warnings

DOI: 10.18129/B9.bioc.DESeq2

Differential gene expression analysis based on the negative binomial distribution

Bioconductor version: Release (3.6)

Estimate variance-mean dependence in count data from high-throughput sequencing assays and test for differential expression based on a model using the negative binomial distribution.

Author: Michael Love, Simon Anders, Wolfgang Huber

Maintainer: Michael Love <michaelisaiahlove at gmail.com>

Citation (from within R, enter `citation("DESeq2")`):

Love MI, Huber W and Anders S (2014). "Moderated estimation of fold change and dispersion for RNA-seq

Differential expression analysis: DESeq2 vignette

```
vignette(DESeq2)
```

Analyzing RNA-seq data with DESeq2

Michael I. Love, Simon Anders, and Wolfgang Huber

11 November 2017

Abstract

A basic task in the analysis of count data from RNA-seq is the detection of differentially expressed genes. The count data are presented as a table which reports, for each sample, the number of sequence fragments that have been assigned to each gene. Analogous data also arise for other assay types, including comparative ChIP-Seq, HiC, shRNA screening, mass spectrometry. An important analysis question is the quantification and statistical inference of systematic changes between conditions, as compared to within-condition variability. The package DESeq2 provides methods to test for differential expression by use of negative binomial generalized linear models; the estimates of dispersion and logarithmic fold changes incorporate data-driven prior distributions This vignette explains the use of the package and demonstrates typical workflows. An RNA-seq workflow on the Bioconductor website covers similar material to this vignette but at a slower pace, including the generation of count matrices from FASTQ files. DESeq2 package version: 1.18.1

- Standard workflow
  - Quick start
  - How to get help for DESeq2
  - Input data
    - Why un-normalized counts?
    - The DESeqDataSet

DESeq and edgeR for DE Analysis

NATURE PROTOCOLS | PROTOCOL

Count-based differential expression analysis of RNA sequencing data using R and Bioconductor

Simon Anders, Davis J McCarthy, Yunshun Chen, Michal Okoniewski, Gordon K Smyth, Wolfgang Huber & Mark D Robinson

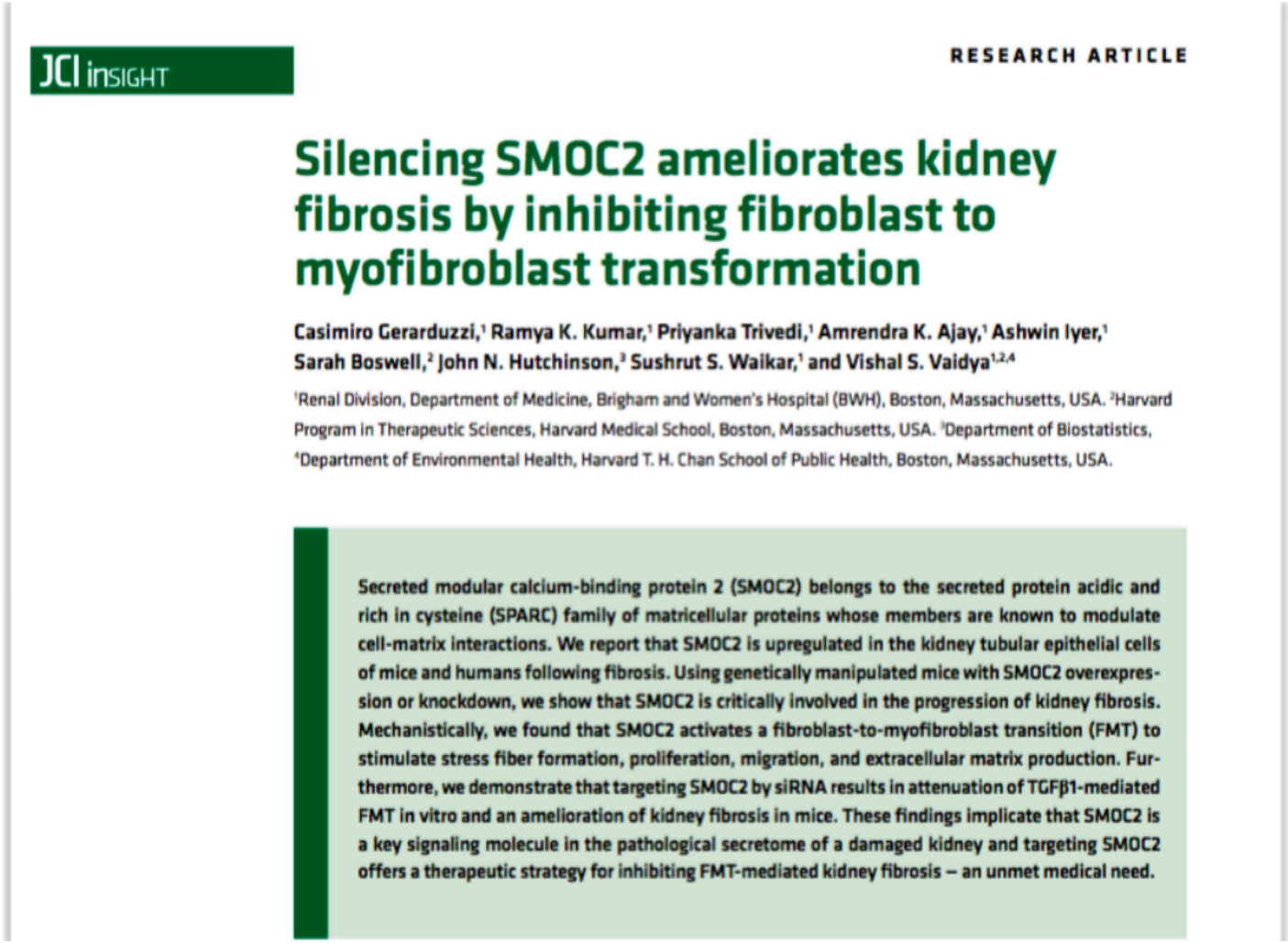
Affiliations | **Contributions** | **Corresponding authors**

Nature Protocols **8**, 1765–1786 (2013) | doi:10.1038/nprot.2013.099
Published online 22 August 2013

Citation | Reprints | Rights & permissions | Article metrics

Objective of hands on

- Independently discover DE genes between your experimental groups



JCI insight

RESEARCH ARTICLE

Silencing SMOC2 ameliorates kidney fibrosis by inhibiting fibroblast to myofibroblast transformation

Casimiro Gerarduzzi,[1] Ramya K. Kumar,[1] Priyanka Trivedi,[1] Amrendra K. Ajay,[1] Ashwin Iyer,[1] Sarah Boswell,[2] John N. Hutchinson,[3] Sushrut S. Waikar,[1] and Vishal S. Vaidya[1,2,4]

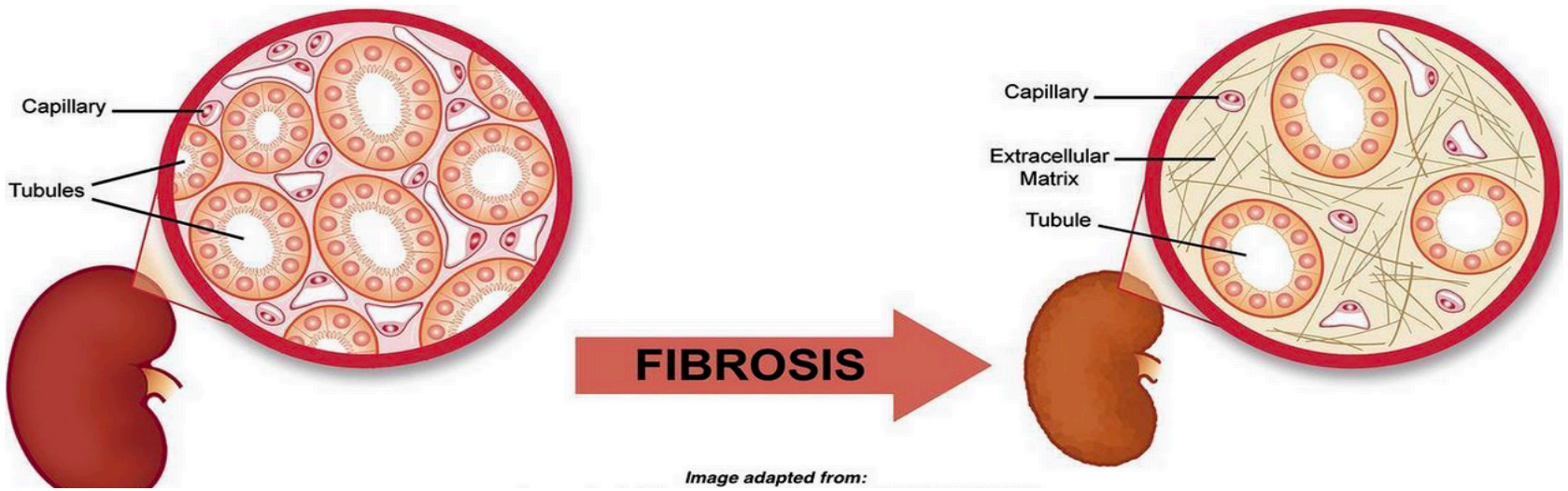
[1]Renal Division, Department of Medicine, Brigham and Women's Hospital (BWH), Boston, Massachusetts, USA. [2]Harvard Program in Therapeutic Sciences, Harvard Medical School, Boston, Massachusetts, USA. [3]Department of Biostatistics, [4]Department of Environmental Health, Harvard T. H. Chan School of Public Health, Boston, Massachusetts, USA.

Secreted modular calcium-binding protein 2 (SMOC2) belongs to the secreted protein acidic and rich in cysteine (SPARC) family of matricellular proteins whose members are known to modulate cell-matrix interactions. We report that SMOC2 is upregulated in the kidney tubular epithelial cells of mice and humans following fibrosis. Using genetically manipulated mice with SMOC2 overexpression or knockdown, we show that SMOC2 is critically involved in the progression of kidney fibrosis. Mechanistically, we found that SMOC2 activates a fibroblast-to-myofibroblast transition (FMT) to stimulate stress fiber formation, proliferation, migration, and extracellular matrix production. Furthermore, we demonstrate that targeting SMOC2 by siRNA results in attenuation of TGFβ1-mediated FMT in vitro and an amelioration of kidney fibrosis in mice. These findings implicate that SMOC2 is a key signaling molecule in the pathological secretome of a damaged kidney and targeting SMOC2 offers a therapeutic strategy for inhibiting FMT-mediated kidney fibrosis – an unmet medical need.

This experiment is designed with SMOC2_oe mice to see if SMOC2 genes are over expressed in oe samples compared to wild type mice.

Over expressed SMOC2 are more likely to develop kidney fibrosis.

Introduction to dataset: Smoc2



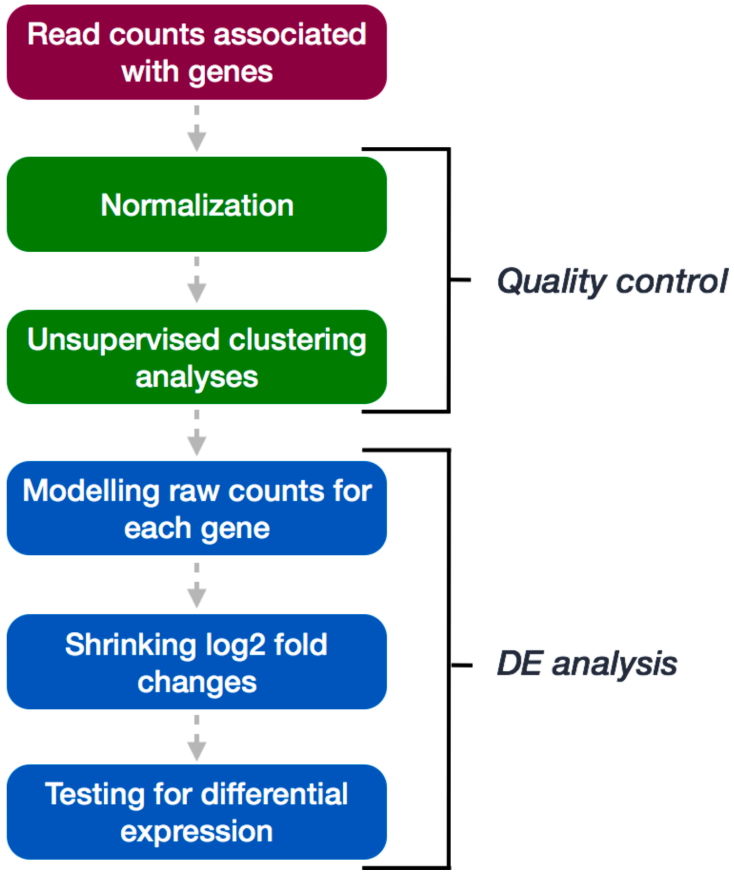
Smoc2 <- secreted modular calcium binding protein 2, shows increase in expression in kidney fibrosis, characterised by increased Extacellular Matrix between Capillary and Tubules in Kidney.

- Four sample groups tested.

Smoc2 Expression

Fibrosis Condition	Wild type	Smoc2 over-expression
Normal	WT normal1	Smoc2 normal1
	WT normal2	Smoc2 normal2
	WT normal3	Smoc2 normal4
Fibrosis	WT fibrosis1	Smoc2 fibrosis1
	WT fibrosis2	Smoc2 fibrosis2
	WT fibrosis3	Smoc2 fibrosis3
	WT fibrosis4	Smoc2 fibrosis4

DESeq2 Workflow



Steps for analysis

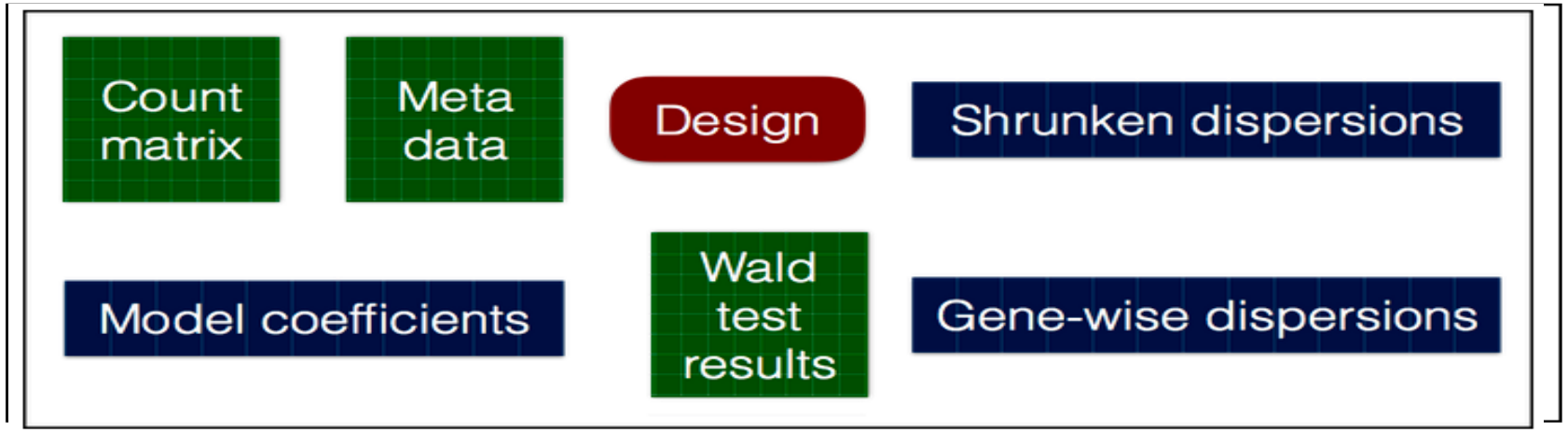
- We are going to use DESeq2 R package for hands on.
- Look at the RNASeq count distribution using ggplot, observe how may genes have 0 counts and how many have high counts.
- Load the raw counts
- Create metadata

	genotype	condition
smoc2_fibrosis1	smoc2_oe	fibrosis
smoc2_fibrosis2	smoc2_oe	fibrosis
smoc2_fibrosis3	smoc2_oe	fibrosis
smoc2_fibrosis4	smoc2_oe	fibrosis
smoc2_normal1	smoc2_oe	normal
smoc2_normal3	smoc2_oe	normal
smoc2_normal4	smoc2_oe	normal

- Qc of raw counts
  - Normalisation to account for library depth variation
  - PCA and Hierarchical clustering of samples (potential sample outliers and major source of variation)
- Creating DESeq2 object
  - Need raw count
  - Meta data
  - Design – which condition/genotype you want to analyse for

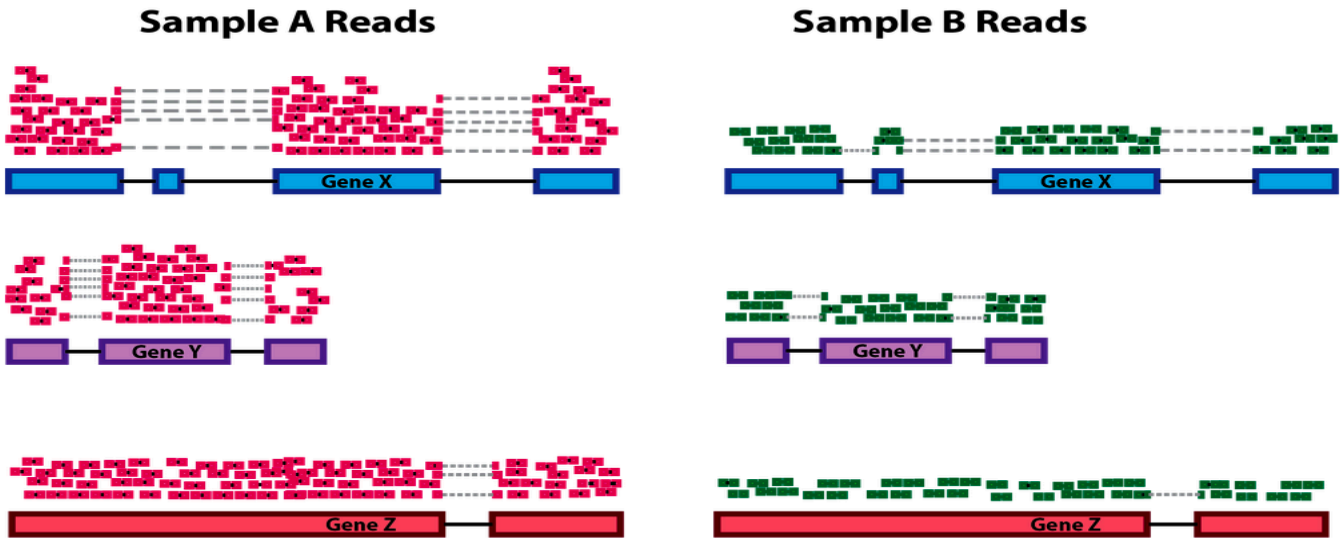
Creating the DESeq2 object

```
# Create DESeq object
dds_wt <- DESeqDataSetFromMatrix(countData = wt_rawcounts,
                                 colData = reordered_wt_metadata,
                                 design = ~ condition)
```

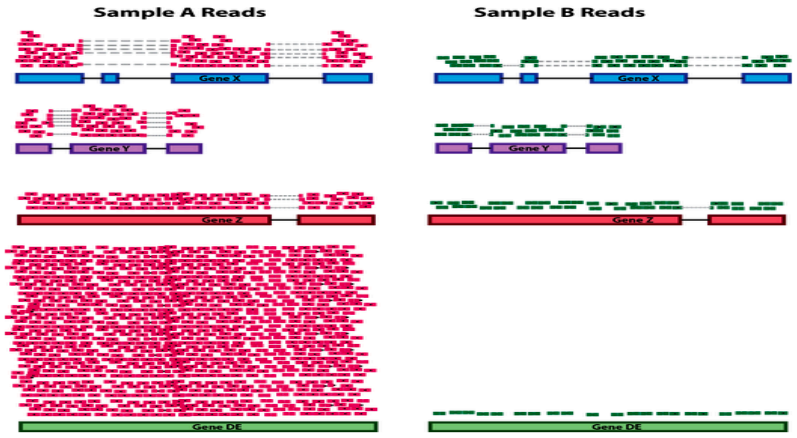


Normalization

Count normalization



Library composition effect



What is normalization: the raw count is number of reads aligned to genes and should be proportional to the amount of RNA in a sample. Although there are number of factors that can affect reads aligned to the genes. We can remove these factor by normalization.

The main factors are library depth, gene length, and RNA composition.

Difference in library size can lead reads map more to genes in one sample than other.

Sample A has 2x reads than B.So need to adjust counts assigned to each gene based on the size of library prior to analysis.

Gene length : long gene has more reads. We don't normalize this since we compare same genes across samples.

Library composition effect: A few highly DEG may skew the normalization method that are not resistant to these outliers. Example green genes takes lot of reads in sample A. divide count by total number of reads will again skew the normalisation. Hence DESeq uses median abrasial normalization for this. This method adjust for library size and resistant for large DEG.

DEG analysis using DESeq2

Major steps

- Fitting the raw counts of each gene to negative binomial model and testing for DE.
- Shrinking log2 fold change
- extracting & visualizing the results

DESeq2 Design Formula

DESeq2 workflow: Design formula

sample	strain	date	cage	treatment	replicate	sex
B1	BALB/cJ	20180515	1	yes	1	M
B2	C57BL/6J	20180515	2	yes	1	M
B3	BALB/cJ	20180515	3	no	1	M
B4	C57BL/6J	20180515	1	no	1	F
B5	BALB/cJ	20180515	2	yes	2	F
B6	C57BL/6J	20180515	3	yes	2	M
B7	BALB/cJ	20180515	1	no	2	M
B8	C57BL/6J	20180515	2	no	2	M
B9	BALB/cJ	20180515	3	yes	3	F
B10	C57BL/6J	20180307	1	yes	3	F
B11	BALB/cJ	20180307	2	no	3	M
B12	C57BL/6J	20180307	3	no	3	M

If all the factors shown to effect DE, then all should be in the design formula
Design = ~sex+strain+treatment+sex:treatment (interaction effect)

Significant DE genes

Significant DE genes - arrange

```
wt_res_sig <- subset(wt_res_all, padj < 0.05)
```

```
wt_res_sig  <- wt_res_sig %>%
        arrange(padj)
```

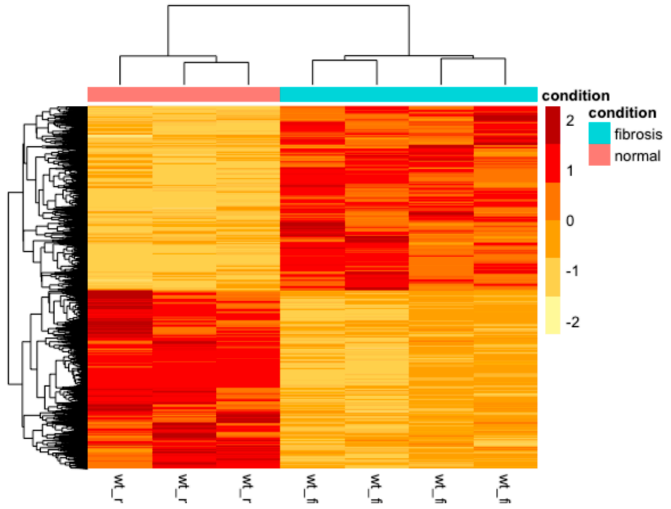
```
View(wt_res_all)
```

ensgene	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	symbol	description
ENSMUSG00000053113	1318.1717	4.875042	0.16021506	28.35016	8.330958e-177	1.830145e-172	Socs3	suppressor of cytokine signaling 3 [Source:M...
ENSMUSG00000005087	2943.7403	6.121134	0.20721978	27.89891	2.750356e-171	3.020991e-167	Cd44	CD44 antigen [Source:MGI Symbol;Acc:MGI:8...
ENSMUSG00000036887	3899.5135	3.866162	0.12740248	27.83465	1.652344e-170	1.209957e-166	C1qa	complement component 1, q subcomponent,...
ENSMUSG00000026822	8870.1712	6.466148	0.23782361	25.82294	4.901029e-147	2.691645e-143	Lcn2	lipocalin 2 [Source:MGI Symbol;Acc:MGI:96757]
ENSMUSG00000036905	3237.6046	3.835279	0.13773926	25.52164	1.134018e-143	4.982421e-140	C1qb	complement component 1, q subcomponent,...
ENSMUSG00000027962	9298.5984	5.781446	0.21949603	24.88019	1.219153e-136	4.463724e-133	Vcam1	vascular cell adhesion molecule 1 [Source:MG...
ENSMUSG00000018008	1278.6520	3.202855	0.11631046	24.77939	1.495690e-135	4.693902e-132	Cyth4	cytohesin 4 [Source:MGI Symbol;Acc:MGI:244...
ENSMUSG00000051439	4144.4097	3.743987	0.14014630	24.43589	7.109040e-132	1.952142e-128	Cd14	CD14 antigen [Source:MGI Symbol;Acc:MGI:8...
ENSMUSG00000019122	1022.6759	6.119309	0.23466958	24.38950	2.210634e-131	5.395911e-128	Ccl9	chemokine (C-C motif) ligand 9 [Source:MGI ...
ENSMUSG00000049103	1459.2660	4.429691	0.17109476	23.99096	3.455388e-127	7.590796e-124	Ccr2	chemokine (C-C motif) receptor 2 [Source:M...
ENSMUSG00000024164	28248.5968	6.095037	0.24563004	23.52525	2.250267e-122	4.493989e-119	C3	complement component 3 [Source:MGI Symb...
ENSMUSG00000022037	50990.1309	3.187649	0.12200514	23.51432	2.911008e-122	5.329085e-119	Clu	clusterin [Source:MGI Symbol;Acc:MGI:88423]
ENSMUSG00000024349	951.8453	3.327356	0.13816314	21.75630	6.021862e-105	1.017602e-101	Tmem173	transmembrane protein 173 [Source:MGI Sym...

Visualizing results

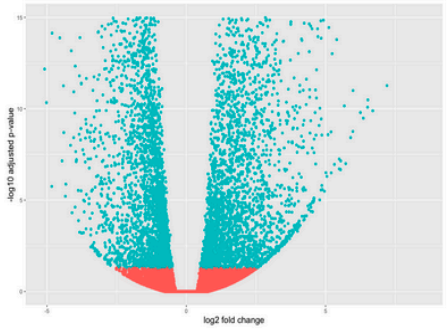
Visualizing results - Expression heatmap

```
# Run pheatmap
pheatmap(sig_norm_counts_wt,
         color = heat_colors,
         cluster_rows = T,
         show_rownames = F,
         annotation = select(wt_metadata, condition),
         scale = "row")
```

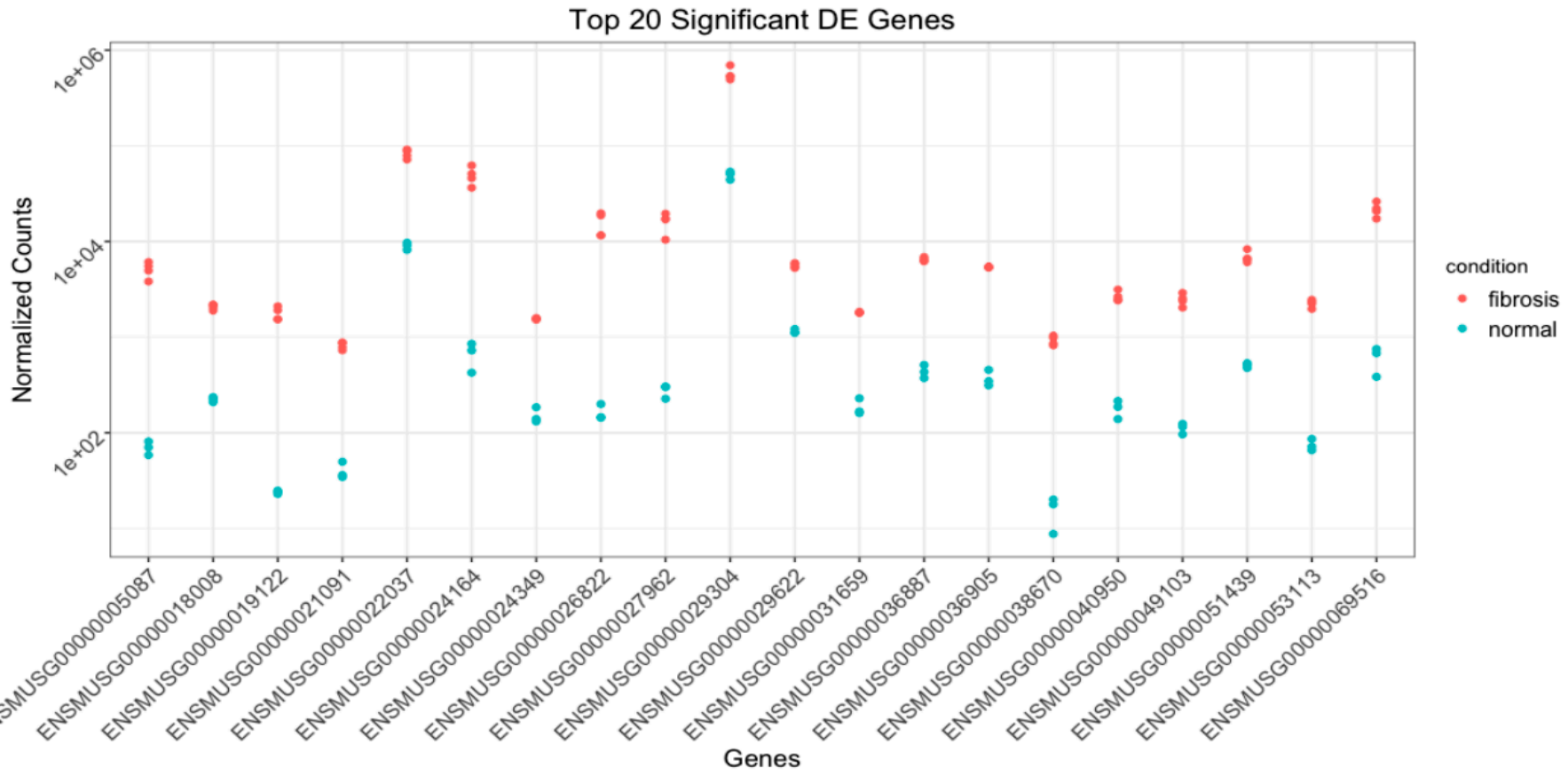


Visualizing results - Volcano plot

```
ggplot(wt_res_all) +
        geom_point(aes(x = log2FoldChange, y = -log10(padj), color = threshol
        xlab("log2 fold change") +
        ylab("-log10 adjusted p-value") +
        ylim=c(0, 15) +
        theme(legend.position = "none",
              plot.title = element_text(size = rel(1.5), hjust = 0.5),
              axis.title = element_text(size = rel(1.25)))
```



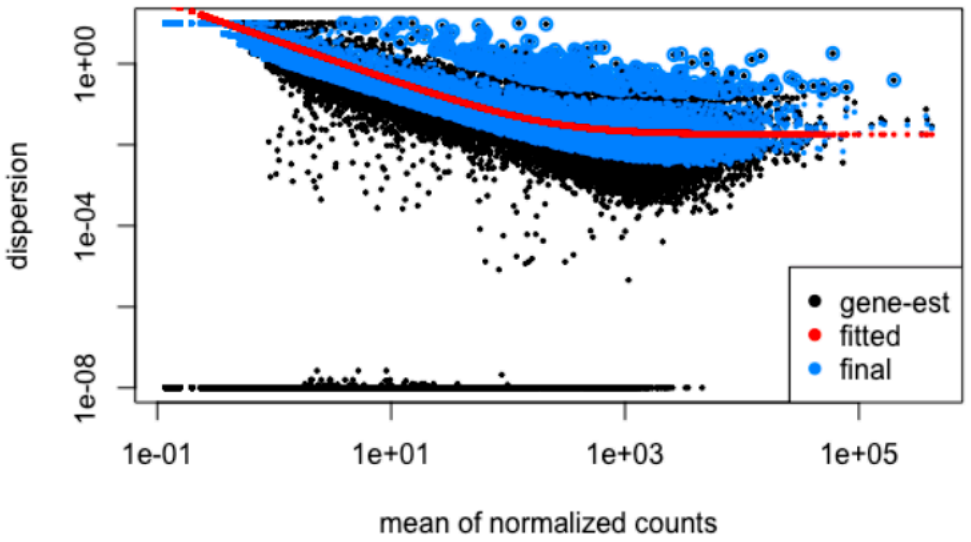
Visualizing results – top 20 genes expression plot



Top 20 genes are up regulated in fibrosis compared to wild type.

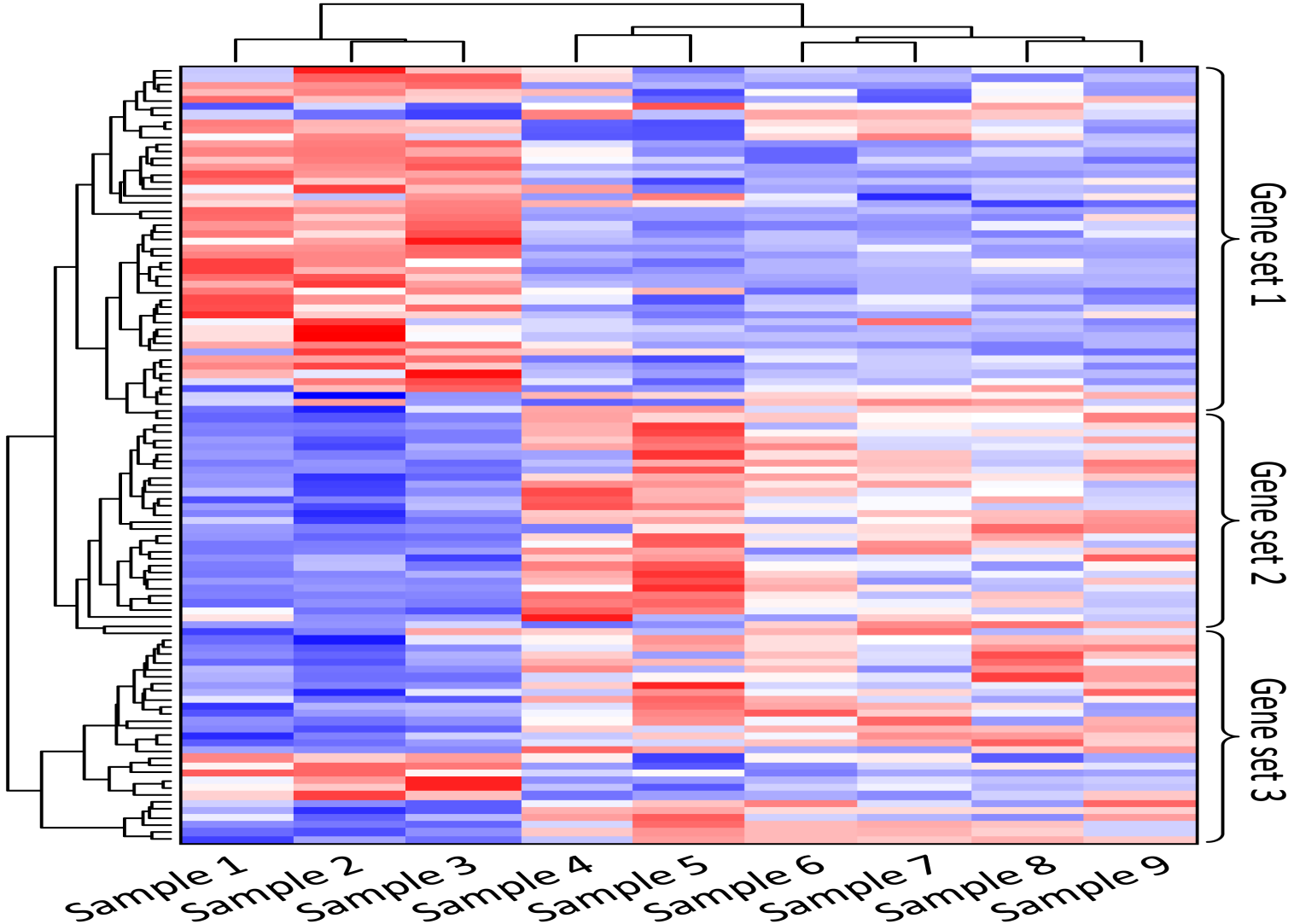
Variance is square of sd – ie how far each value is from their mean
For RNASeq variance is expected to increase with increase in gene's mean expression.
Measure of variance for a given mean is measured using dispersion which used to model variability in expression.
Dispersion indirectly related to the mean and directly to variance.
Variance = mean + dispersion * (mean*mean)

- **Relationship between mean, variance and dispersion:**
- ↑variance⇒↑dispersion
- ↑mean⇒↓dispersion
- For any two gene with same mean expression, the only difference in dispersion will be based on difference in variance.
- Lower alpha indicates less probability of identifying DEG as DE when actually they are not.
- Contrasts is list of samples to be compared



Each dot is gene with mean expression and dispersion. As mean increases, dispersion decreases.
The original dispersion model black curve is shrunken to fit the red line.
Larger number of replicates may fit more and do not need lfc shrinkage.

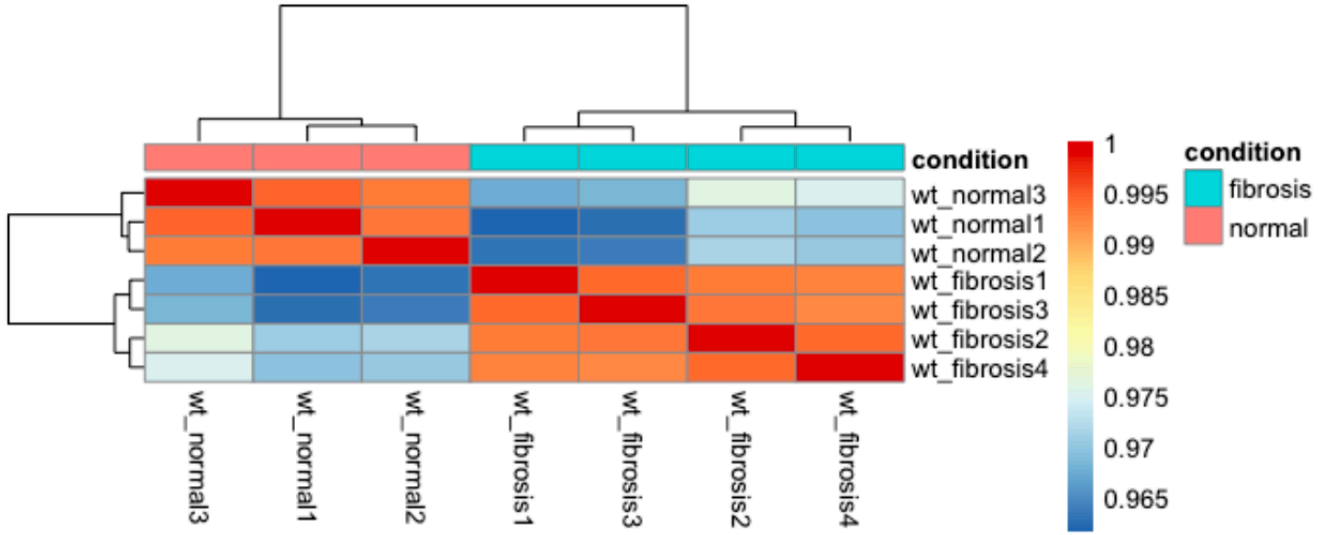
DE of genes among samples

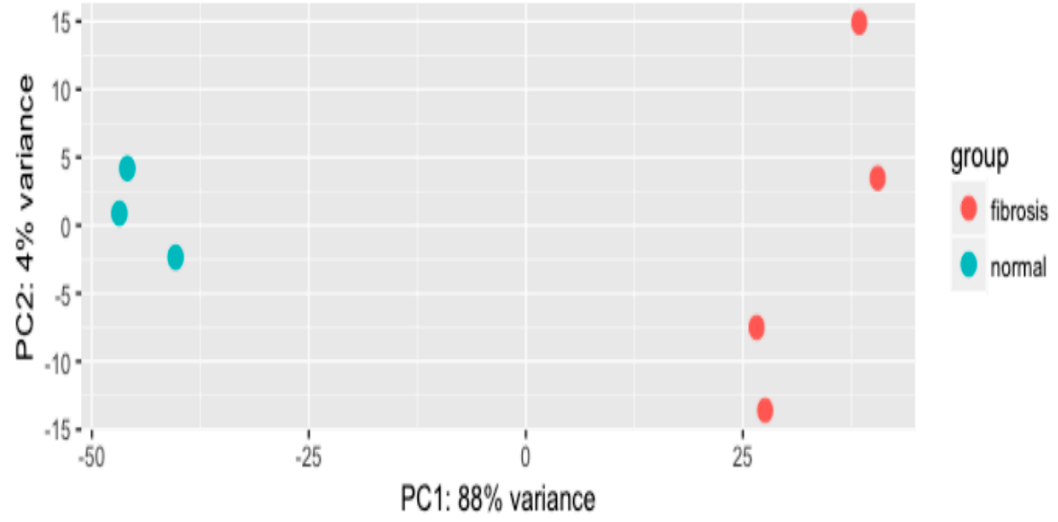


Unsupervised clustering of samples

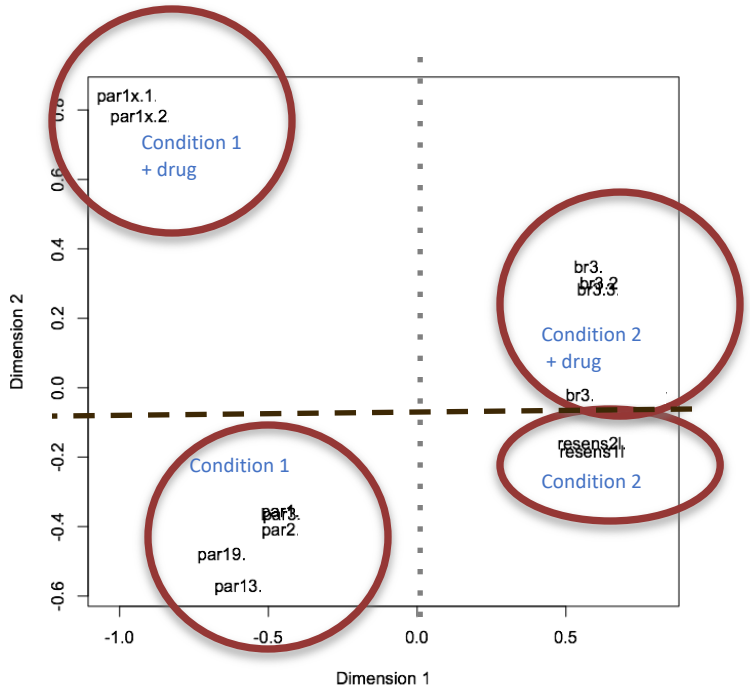
- how similar are the samples to each other with regards to gene expression and identify outliers
- Log transform the normalized counts to improve the visualization of clustering (In DESeq2 vst variant stabilising transformatin is used, blind option is to ignore sample information)
- Biological replicates should cluster together, while samples apart with all correlation factors high. If you find an outlier in both Heatmap and PCA, consider to remove from the analysis.

Hierarchical clustering with correlation heatmaps

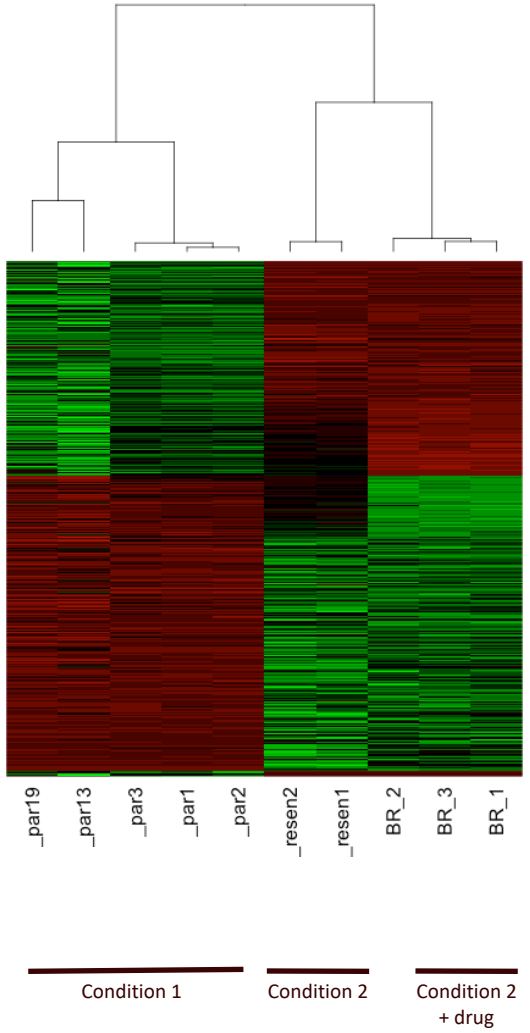




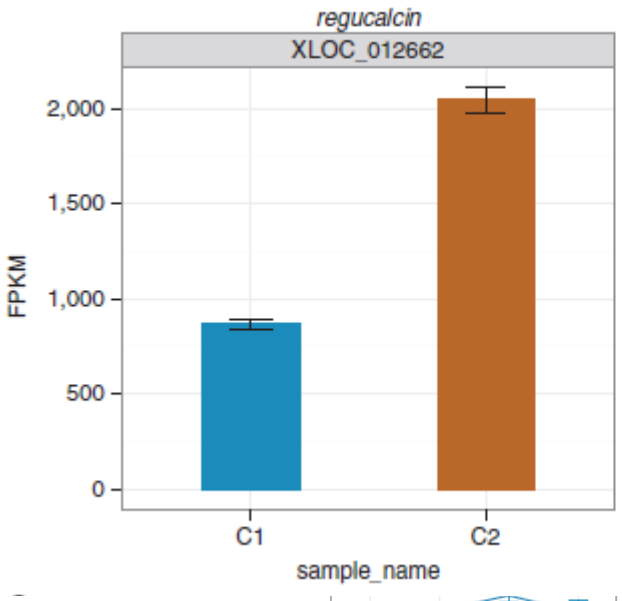
RNAseq post-analysis & QC
PCA plots & corresponding heatmap

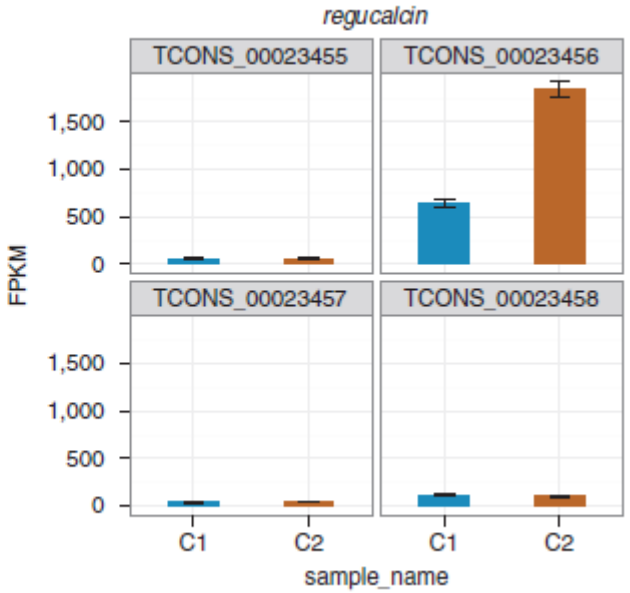


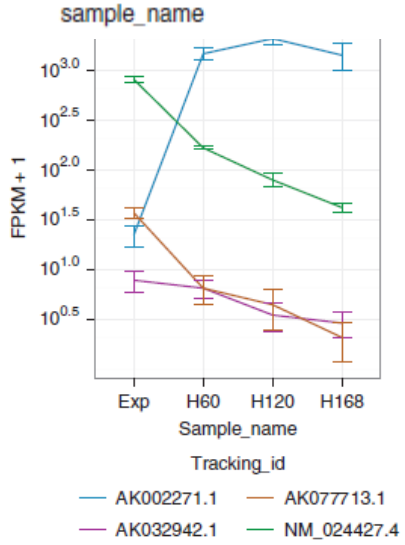
PCA plot separates:
(1) Condition 1 vs. Condition 2
(2) Lines grown in drug or normal media

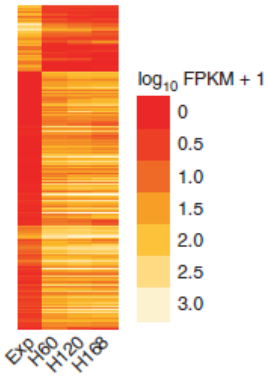


CummeRbund


regucalcin
XLOC_012662
FPKM
2,000
1,500
1,000
500
0
C1
C2
sample_name


regucalcin
TCONS_00023455
TCONS_00023456
TCONS_00023457
TCONS_00023458
FPKM
1,500
1,000
500
0
C1
C2
sample_name


FPKM + 1
$10^{3.0}$
$10^{2.5}$
$10^{2.0}$
$10^{1.5}$
$10^{1.0}$
$10^{0.5}$
Exp
H60
H120
H168
Sample_name
Tracking_id
AK002271.1
AK077713.1
AK032942.1
NM_024427.4


$\log_{10}$ FPKM + 1
0
0.5
1.0
1.5
2.0
2.5
3.0
Exp
H60
H120
H168

Hands on differential expression

1. Let us start with Rstudio
2. Make sure you set the working directory to the folder with count table. Setwd("pathtocounttables")
3. Create a new R script file and assign a name to it
4. Load the R libraries needed

```
library("DESeq2")
library(ggplot2)
library("RColorBrewer")
library("pheatmap")
library("vsn")
library(apeglm)
library(annotables)
library(biobroom)
library(dplyr)
library(ggrepel)
library("EnhancedVolcano")
```

5. Load the count file

```
raw_data <- read.csv("fibrosis_counts.csv", header=TRUE, row.names = 1)

class(raw_data)

dim(raw_data)

head(raw_data)
```

7. Select either normal or UUO data

```
#selected_data <- raw_data[, c( 11:14,4:7)] ## to get wt_UUO and Smoc_UUO
#selected_data <- raw_data[, c( 8:10,1:3)] ## to get wt_norm and Smoc_norm
### count table from paper
selected_data <- raw_data[, c(1:3, 8:10)] ### to get wt_norm and smoc_norm
selected_data <- as.matrix(selected_data)
dim(selected_data)
class(selected_data)
head(selected_data)
```

8.

```
 ###Before normalisation counts plot
dat_log2 <- stack(as.data.frame(log2(selected_data)))
ggplot(data = dat_log2, mapping = aes(x = ind, y = values)) +
  geom_jitter(alpha = 0.3, color = "tomato") +
  geom_boxplot(alpha = 0)
##Expt meta data ----
##Create the DESeq2 data object
#genotype <- c(rep("WT", 4),rep("TRTED", 4))  ### for UUO
genotype <- c(rep("WT", 3),rep("TRTED", 3))
coldata <- data.frame(genotype)
rownames(coldata) <- colnames(selected_data)
coldata

dds <- DESeqDataSetFromMatrix(countData = selected_data,
                              colData = coldata,
                              design = ~ genotype)
dds
summary(dds)
```

9.

```
###
## 0.2 Pre-filtering ----
keep <- rowSums(counts(dds)) >= 0 # 0 means no pre-filtering
dds <- dds[keep,]

dds$genotype <- factor(dds$genotype, levels = c("WT", "TRTED"))
dds$genotype


###Normalization ----
dds <- estimateSizeFactors(dds)
normalized_data <- counts(dds, normalized=TRUE)
normalized_data

### visualize normalized data
dat_log2 <- stack(as.data.frame(log2(normalized_data)))
###Stacking vectors concatenates multiple vectors into a single vector along with a factor indicating where
#each observation originated. Unstacking reverses this operation. (from utils package)

ggplot(data = dat_log2, mapping = aes(x = ind, y = values)) +
  geom_jitter(alpha = 0.3, color = "forestgreen") +
  geom_boxplot(alpha = 0)
  #geom_jitter(alpha = 0.3, color = "darkred") +  #color = gold1, tomato, forestgreeen
```

10. Clustering of samples

```
#### Clustering of samples ----
#Unsupervised clustering of samples
#Result: Heatmap of all sample distance & PCA plot

rld <- rlog(dds, blind=FALSE) # Extracting transformed values

vsd <- vst(dds, blind=FALSE)

sampleDists <- dist(t(assay(rld)))
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- rld$genotype
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
         clustering_distance_rows=sampleDists,
         clustering_distance_cols=sampleDists,
         col=colors)
```

11. PCA plot of samples

```
### Variance stabilised transformed
sampleDists <- dist(t(assay(vsd)))

sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- rld$genotype
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
         clustering_distance_rows=sampleDists,
         clustering_distance_cols=sampleDists,
         col=colors)
###dimensionality reduction of samples
# of rlog values
plotPCA(rld, intgroup=c("genotype"))
#of vst values
plotPCA(vsd, intgroup=c("genotype"))
```

12. DE Analysis

```
### RUN DE Analysis
#Run DESeq2
dds <- DESeq(dds)
structure(dds)
resultsNames(dds)
res <- results(dds)
res

### Mean - Variance relationship
#Mean-variance relationship of wild type and smoc samples
#Result: Mean Vs variance plot
ntd <- normTransform(dds)
meanSdPlot(assay(ntd))
meanSdPlot(assay(vsd))
meanSdPlot(assay(rld))
## Now mean-variance plot after DE using DESeq2
plotDispEsts(dds)
```

13. continued

```
### MA plot
#Differential expression across conditions.
#MA Plots before log fold change shrinkage
plotMA(res, ylim=c(-5,5))


###MA Plots after log fold change shrinkage
resLFC <- lfcShrink(dds, coef = "genotype_TRTED_vs_WT", type = "apeglm")
plotMA(resLFC, ylim=c(-4, 4))

###Filter significant genes for alpha=0.05 and fold change 1.25
#Results: number of genes before filtering
summary(res)

##Results: number of genes before filtering with > 1.25 fold change
res_fc_1.25 <- results(dds, lfcThreshold = log2(1.25)) # log2(1.25))
summary(res_fc_1.25)

#Results: number of genes after filtering: > 1.25 fold change & padj < 0.05
res_fc_1.25_alpha_0.05 <- results(dds, lfcThreshold=log2(1.25), alpha=0.05)
summary(res_fc_1.25_alpha_0.05)
```

14. Add annotation to your gene ids

```
###Add ensembl annotation using annotables R package to filtered genes
#Result: table of filtered genes with annotation
resSig <- subset(res_fc_1.25_alpha_0.05, padj < 0.05)
summary(resSig)
#, lfcThreshold=log2(1.25), alpha=0.05)
resSig
####Annotation ## not needed if genes are symbols (NOT ensembleid)
library(annotables)


resSig_tidy <- tidy.DESeqResults(resSig)

resSig_tidy %>%
  dplyr::arrange(p.adjusted) %>%
  dplyr::inner_join(grcm38, by = c("gene" = "ensgene")) %>%
  dplyr::select(gene, estimate, p.adjusted, symbol) %>%
  knitr::kable(.)

# with description

resSig_tidy %>%
  dplyr::arrange(p.adjusted) %>%
  dplyr::inner_join(grcm38, by = c("gene" = "ensgene")) %>%
  dplyr::select(gene, estimate, p.adjusted, symbol, description) %>%
  knitr::kable(.)
```

15. Get Volcano plot of significant genes

```
####Get subset of normalized significant genes with padj < 0.05
#Result: table of those genes and heatmap of them across conditions

df <- as.data.frame(colData(dds)[,c("genotype")])
colnames(df) <- c("genotype")
rownames(df) <- colnames(vsd)

mat = assay(vsd)[ head(order(res$padj), 20), ] # select the top 20 genes with the lowest padj
mat = mat - rowMeans(mat)

pheatmap(mat, cluster_rows=TRUE, show_rownames=TRUE,
         cluster_cols=TRUE, annotation_col=df)

###Volcano plot of significant genes from step 10 Result: Volcano plot
#Result: Volcano plot on all genes
EnhancedVolcano(res,
                lab = rownames(res),
                x = 'log2FoldChange',
                y = 'pvalue',
                xlim = c(-10, 10),
                selectLab = c(""))
###Result: Volcano plot on significant genes
EnhancedVolcano(resSig,
                lab = rownames(resSig),
                x = 'log2FoldChange',
                y = 'pvalue',
                xlim = c(-10, 10),
                selectLab = c(""))
```

16. Expression plot of top 20 genes

```
####Expression plot of top 20 genes (least padj) across samples
#Result: Expression plot
resSig_tidy_top_20 <- resSig_tidy %>%
  dplyr::arrange(p.adjusted) %>%
  head(20) %>%
  dplyr::inner_join(grcm38, by = c("gene" = "ensgene")) %>%
  dplyr::select(gene, estimate, p.adjusted, symbol)

x_label <- c(as.matrix(resSig_tidy_top_20[, "symbol"]))
x_label

# resSig_tidy_top_20 <- as.data.frame(res) %>% tibble::rownames_to_column("gene")%>%
#   dplyr::arrange(padj) %>%
#   head(19) %>% dplyr::select(gene, log2FoldChange, padj)
#   #dplyr::inner_join(grcm38, by = c("gene" = "ensgene")) %>%
#   #dplyr::select(gene, estimate, p.adjusted, symbol)
#x_label <- c(as.matrix(resSig_tidy_top_20[, "gene"]))
#x_label

###plotting top 20 genes across all samples
resSig
resSigOrdered <- resSig[order(resSig$padj),][1:20,]    #1:20 for larger sig genes
exp_plot_data <- as.data.frame(assay(vsd)[rownames(resSigOrdered), ])
colnames(exp_plot_data) <- coldata[, "genotype"]
```

17. Continued

```
class(exp_plot_data)


library(reshape2)

exp_plot_data_matrix <- cbind(ID=rownames(exp_plot_data),  exp_plot_data)
exp_plot_data_matrix <- melt(exp_plot_data_matrix)

ggplot(data = exp_plot_data_matrix, mapping = aes(x = ID, y = value, color = variable)) +
  geom_jitter(alpha = 0.3, width = 0.25) +
  scale_y_log10() +
  theme(axis.text.x = element_text(colour = "grey20", size = 8, angle = 60, hjust = 1.0, vjust = 1.0),
        axis.text.y = element_text(colour = "grey20", size = 12),
        text = element_text(size = 16)) +
  scale_x_discrete(labels=x_label) +
  xlab("Gene Symbol") +
  ylab("Normalized expression value")

ggplot(data = exp_plot_data_matrix, mapping = aes(x = ID, y = value, color = variable)) +
  geom_jitter(alpha = 0.3, width = 0.25) +
  scale_y_log10() +
  theme(axis.text.x = element_text(colour = "grey20", size = 8, angle = 60, hjust = 1.0, vjust = 1.0),
        axis.text.y = element_text(colour = "grey20", size = 12),
        text = element_text(size = 16)) +
  xlab("Ensembl ID") +
  ylab("Normalized expression value")


res_nona <- na.omit(res)
dim(res_nona)
write.table(res_nona, file="DE_genes_all.txt", sep="\t")
```

Functional Analysis of DE Genes and Alternative splicing

By Dr. Alaguraj Veluchamy


THANK YOU!
QUESTIONS?