RNA sequence practical

Michael Inouye
Baker Heart and Diabetes Institute
Univ of Melbourne / Monash Univ

Summer Institute in Statistical Genetics 2017
Integrative Genomics Module
Seattle

@minouye271
www.inouyelab.org

What are we going to do?

- RNAseq data files
- Align fastq's to a reference
- Quantify gene expression
- Explore the data
- Perform a differential gene expression analysis
- This tutorial makes no assumptions about proficiency in programming or R

What you need

- **Software**
  - **R** (v3.0 or above)
    - **Rsubread** package
      - Enter the following into R:

```
source("http://bioconductor.org/biocLite.R")
biocLite(”Rsubread")
```

    - **edgeR** package
      - Enter the following into R:

```
source("http://bioconductor.org/biocLite.R")
biocLite("edgeR")
```

- **Data**
  - **Mouse genome (mm10)**

http://hgdownload.cse.ucsc.edu/goldenpath/mm10/chromosomes/**chr1.fa**

Load Rsubread and set working directory

```
library(Rsubread)

setwd("whatever_dir_your_data_is_in")
```

What is Rsubread doing?

- **Alignment of reads to a reference genome**
  - The first step in many genomics analyses

```
Read:      GACTGGGCGATCTCGACTTCG
           |||||  |||||||||| |||
Reference: GACTG--CGATCTCGACATCG
```

- **RNAseq – can be used to align to splice junctions (same as many other aligners)**

- **Aligned read counts at genomic features then become a quantitative measure of the expression**

Rsubread: Building an index

- **Usage (will take a minute or 2):**

```
buildindex(basename="reference_index",reference="chr1.fa")
```

<reference>	Reference genome in FASTA format
<basename>	The basename for output index files

- **Why does an aligner need an index?**
  - Alignment is basically a search algorithm
  - Indexing is a way of parsing and storing data to enable accurate and rapid retrieval
  - Align millions of reads very quickly and with low memory
  - Human genome fits into 2.2Gb on disk with ~1.3Gb memory

Alignment

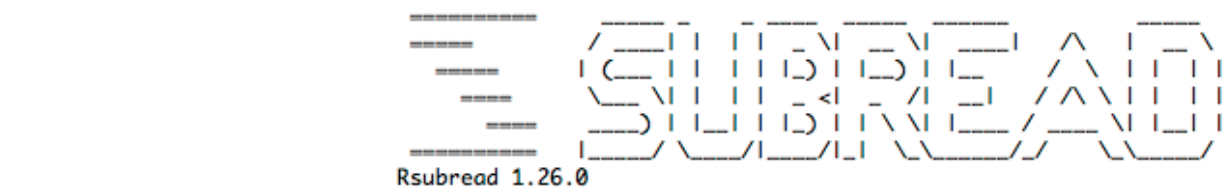
- **Usage (for each pair of read files):**

```
align(index="reference_index", readfile1="sub_chr1_CellStateA_replicate1_1.fq",
readfile2="sub_chr1_CellStateA_replicate1_2.fq", output_file="sub_chr1_CellStateA_replicate1_PE.SAM")
```

-index	Basename for ref index file
-readfile1	File for fastq for one end of paired end sequencing
-readfile2	Other end of PE sequencing
-output_file	Output file in SAM format (doesn't automatically append '.sam')

- Try typing **?align**
  - This prints out a help text

Rsubread: STDERR



```
Rsubread 1.26.0

//========================== featureCounts setting ===========================\\
||                                                                            ||
||            Input files : 1 BAM file                                        ||
||                          P sub_chr1_CellStateB_replicate1_PE.SAM           ||
||                                                                            ||
||      Dir for temp files : .                                                ||
||                 Threads : 1                                                ||
||                   Level : meta-feature level                               ||
||               Paired-end : yes                                             ||
||          Strand specific : no                                              ||
||       Multimapping reads : not counted                                     ||
|| Multi-overlapping reads : not counted                                      ||
||   Min overlapping bases : 1                                                ||
||                                                                            ||
||          Chimeric reads : counted                                          ||
||        Both ends mapped : not required                                     ||
||                                                                            ||
\\====================== http://subread.sourceforge.net/ ======================//

//================================ Running ===================================\\
||                                                                            ||
|| Load annotation file ./.Rsubread_UserProvidedAnnotation_pid13675 ...       ||
||    Features : 222996                                                       ||
||    Meta-features : 27179                                                   ||
||    Chromosomes/contigs : 43                                                ||
||                                                                            ||
|| Process BAM file sub_chr1_CellStateB_replicate1_PE.SAM...                  ||
||    Paired-end reads are included.                                          ||
||    Assign fragments (read pairs) to features...                            ||
||    Total fragments : 104877                                                ||
||    Successfully assigned fragments : 31758 (30.3%)                         ||
||    Running time : 0.01 minutes                                             ||
||                                                                            ||
||                         Read assignment finished.                          ||
||                                                                            ||
\\====================== http://subread.sourceforge.net/ ======================//
```

Load mouse genome (mm10) annotation

- Thankfully it's built in here

- Command

```
mm10 <- getInBuiltAnnotation("mm10")
mm10[1:5,]
```

```
fc_A1 <- featureCounts("sub_chr1_CellStateA_replicate1_PE.SAM",annot.ext=mm10,isPairedEnd=TRUE)
fc_A2 <- featureCounts("sub_chr1_CellStateA_replicate2_PE.SAM",annot.ext=mm10,isPairedEnd=TRUE)
fc_B1 <- featureCounts("sub_chr1_CellStateB_replicate1_PE.SAM",annot.ext=mm10,isPairedEnd=TRUE)
fc_B2 <- featureCounts("sub_chr1_CellStateB_replicate2_PE.SAM",annot.ext=mm10,isPairedEnd=TRUE)
```

Count reads at genomic features

- **Many tools for this**
  - Here we use generic featureCounts

- **What do we need? What features are we looking at?**
- **Command**

```
fc_A1 <- featureCounts("sub_chr1_CellStateA_replicate1_PE.SAM",annot.ext=mm10,isPairedEnd=TRUE)
fc_A2 <- featureCounts("sub_chr1_CellStateA_replicate2_PE.SAM",annot.ext=mm10,isPairedEnd=TRUE)
fc_B1 <- featureCounts("sub_chr1_CellStateB_replicate1_PE.SAM",annot.ext=mm10,isPairedEnd=TRUE)
fc_B2 <- featureCounts("sub_chr1_CellStateB_replicate2_PE.SAM",annot.ext=mm10,isPairedEnd=TRUE)
fc_A1$stat
fc_A2$stat
fc_B1$stat
fc_B2$stat
```

Tabulate counts, load edgeR, etc…

```
D <- cbind(fc_A1$counts, fc_A2$counts, fc_B1$counts, fc_B2$counts)
```

```
source("http://bioconductor.org/biocLite.R")
biocLite("edgeR")
library(edgeR)
```

```
g <- c("stateA","stateA","stateB","stateB")
```

Trim lowly expressed exons

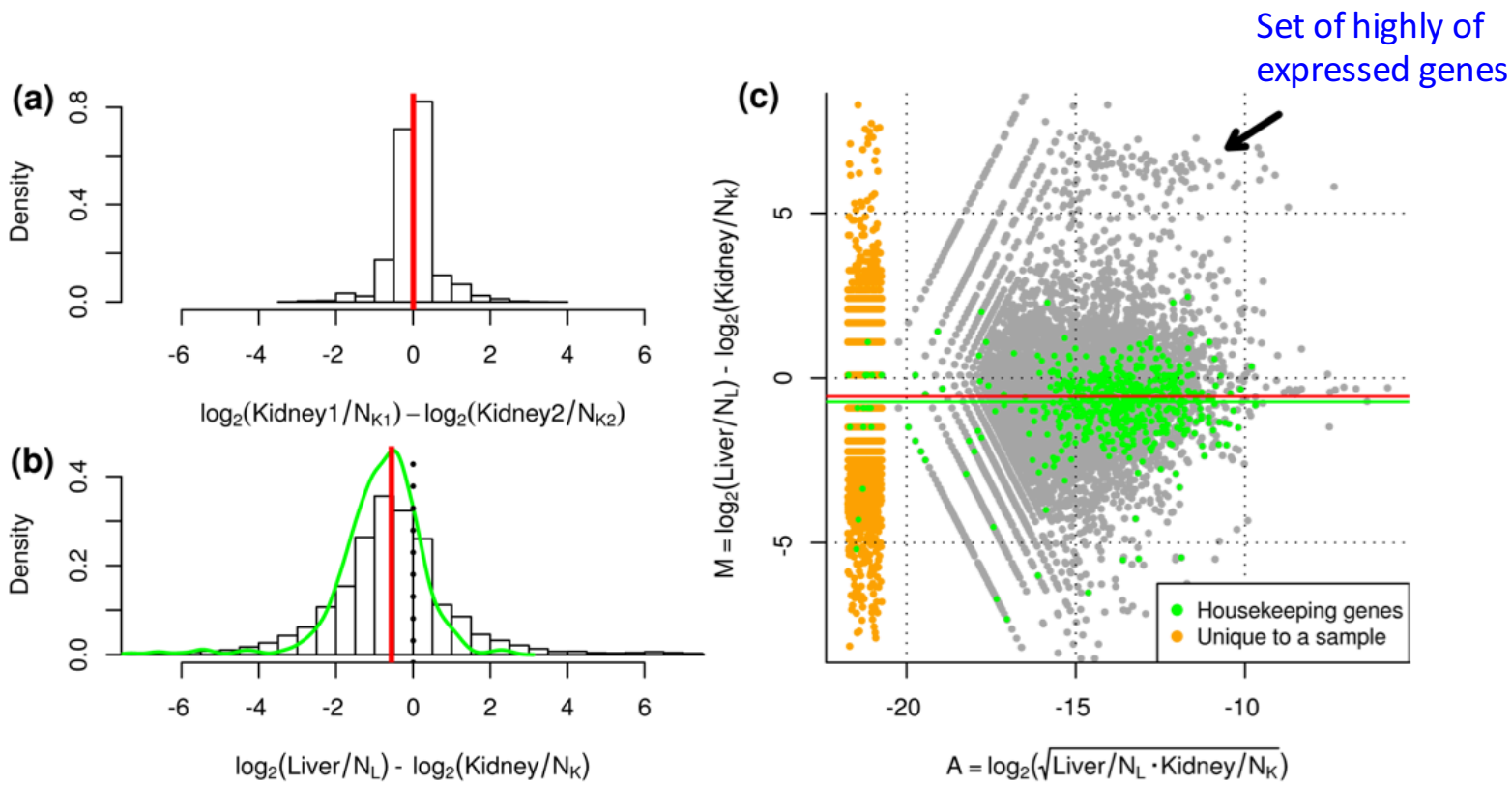
- **Command**

```
D2 <- DGEList(counts=D[,1:4], group=g)
dim(D2)
keep <- rowSums(cpm(D2)>1) >= 1
D2 <- D2[keep,]
dim(D2)
```

TMM normalisation

- **Command**

```
d <- calcNormFactors(D2, method="TMM")
```



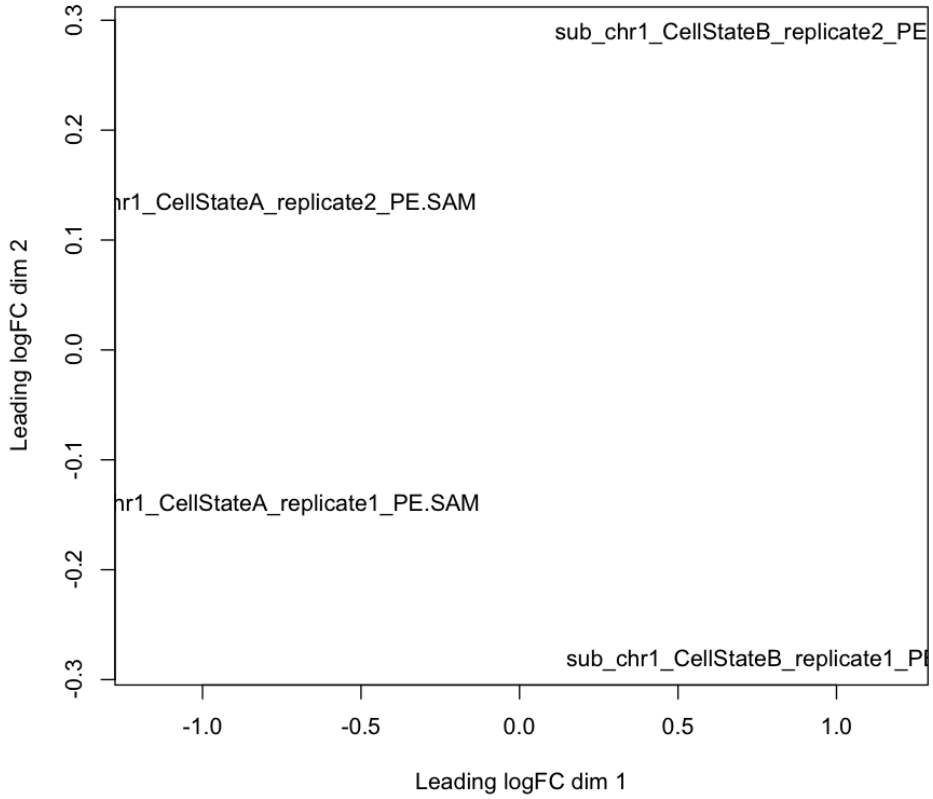
Robinson & Oshlack, *Genome Biology* 2010

Data exploration

- **Command**

  plotMDS(d)

- **Distance b/n a pair of samples**
  - Root mean square deviation (Euclidean distance) for the top 500 genes in terms of fold-change



Estimate common dispersion

- What is the overall biological coefficient of variation in the data?

- **Command**

```
d <- estimateCommonDisp(d, verbose=TRUE)
```

- **Output: Dispersion parameter and Biological Coefficient of Variation**
  - How are these interpreted?
  - How else can we estimate dispersion?

Differential expression

- **Gene by gene, test for differences in means between 2 groups of NB distributed counts**
  - **Command**

```
de_exact <- exactTest(d)
```

- **What are the top DE genes?**
  - **Command**

```
top_genes <- topTags(de_exact, n=25,
    adjust.method="BH", sort.by="PValue")
```

View the topTags

- **What are we looking at?**

```
Comparison of groups:  stateB-stateA
              logFC   logCPM        PValue           FDR
12902     -8.548788 13.44112 9.633143e-254 7.918444e-251
69169     -3.762800 14.80794 3.334798e-222 1.370602e-219
20343     -5.528008 13.33396 3.112540e-205 8.528360e-203
13136     -3.898677 12.85733 4.143449e-126 8.514787e-124
75345      3.348831 12.83569 2.530149e-114 4.159565e-112
236312    -4.618018 12.22352 3.809524e-110 5.219048e-108
110611     2.653711 13.62987 7.523083e-107 8.834249e-105
98267     -2.708680 13.42357  1.554370e-98  1.597115e-96
12487      8.439833 11.24447  9.250587e-97  8.448869e-95
19735     -5.751132 11.49098  1.780790e-86  1.463810e-84
15950     -2.596824 12.87137  2.338490e-76  1.747490e-74
19264     -1.884974 14.54754  1.652764e-72  1.132143e-70
16565     -2.348044 13.05331  4.761744e-70  3.010887e-68
107508     2.438526 12.64620  5.574587e-68  3.273079e-66
240921    -6.044939 11.02242  1.390870e-67  7.621967e-66
13849     -4.499629 11.25451  2.015078e-65  1.035246e-63
100040462 -2.142394 13.08198  6.496215e-61  3.141111e-59
214854    -6.686469 10.74559  2.566532e-58  1.172049e-56
98752     -2.081629 12.95493  9.668748e-56  4.183006e-54
100033459 -7.388281 10.59004  3.355467e-55  1.379097e-53
320011     2.053522 12.60279  3.270406e-51  1.280131e-49
226594    -2.190454 12.50870  4.983638e-51  1.862068e-49
17215      2.400329 11.87671  1.840964e-48  6.579444e-47
72265      1.784103 13.06640  2.040388e-47  6.988330e-46
```

How many genes are differentially expressed?

- FDR 5%

- **Command**

```
summary(de <- decideTestsDGE(de_exact), adjust.method="BH",
p.value=0.05)
```

- **Output (down, NS, up)**

```
-1   151
0    521
1    150
```

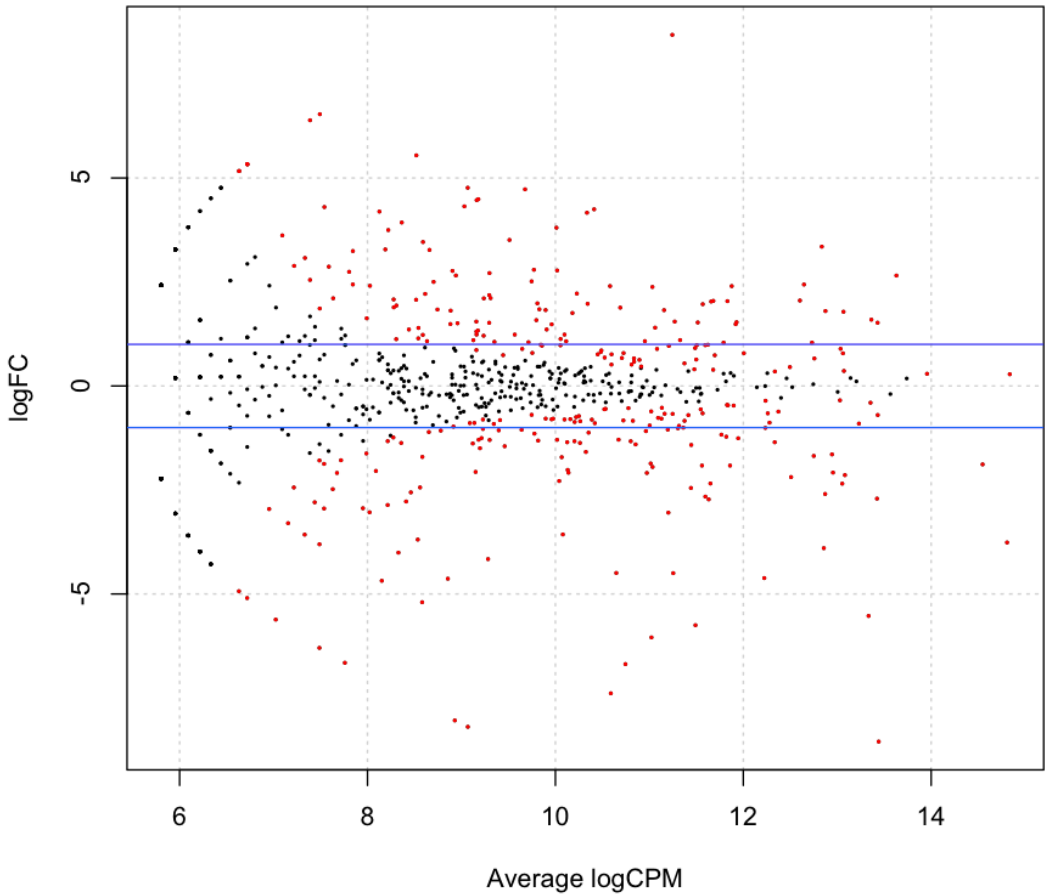
Smear plot

- **Command**

```
detags <-
rownames(d)[as.logical(de)]

plotSmear(de_exact,
de.tags=detags)

abline(h=c(-1, 1), col="blue")
```



Write output for further processing

- **Command**

```
write.table(topTags(de_exact,n=1000000)$table, file=[output])
```

All the key R commands

```
source("https://bioconductor.org/biocLite.R")
biocLite("Rsubread")
library(Rsubread)
setwd("")
buildindex(basename="reference_index",reference="chr1.fa")
align(index="reference_index", readfile1="sub_chr1_CellStateA_replicate1_1.fq", readfile2="sub_chr1_CellStateA_replicate1_2.fq",
output_file="sub_chr1_CellStateA_replicate1_PE.SAM")
align(index="reference_index", readfile1="sub_chr1_CellStateA_replicate2_1.fq", readfile2="sub_chr1_CellStateA_replicate2_2.fq",
output_file="sub_chr1_CellStateA_replicate2_PE.SAM")
align(index="reference_index", readfile1="sub_chr1_CellStateB_replicate1_1.fq", readfile2="sub_chr1_CellStateB_replicate1_2.fq",
output_file="sub_chr1_CellStateB_replicate1_PE.SAM”)
align(index="reference_index", readfile1="sub_chr1_CellStateB_replicate2_1.fq", readfile2="sub_chr1_CellStateB_replicate2_2.fq",
output_file="sub_chr1_CellStateB_replicate2_PE.SAM")
mm10 <- getInBuiltAnnotation("mm10")
fc_A1 <- featureCounts("sub_chr1_CellStateA_replicate1_PE.SAM", annot.ext=mm10,isPairedEnd=TRUE)
fc_A2 <- featureCounts("sub_chr1_CellStateA_replicate2_PE.SAM", annot.ext=mm10,isPairedEnd=TRUE)
fc_B1 <- featureCounts("sub_chr1_CellStateB_replicate1_PE.SAM",annot.ext=mm10,isPairedEnd=TRUE)
fc_B2 <- featureCounts("sub_chr1_CellStateB_replicate2_PE.SAM",annot.ext=mm10,isPairedEnd=TRUE)
D <- cbind(fc_A1$counts, fc_A2$counts, fc_B1$counts, fc_B2$counts)
source("http://bioconductor.org/biocLite.R")
biocLite("edgeR")
library(edgeR)
g <- c("stateA","stateA","stateB","stateB")
D2 <- DGEList(counts=D[,1:4], group=g)
keep <- rowSums(cpm(D2)>1) >= 1
D2 <- D2[keep,]
d <- calcNormFactors(D2, method="TMM")
d <- estimateCommonDisp(d, verbose=TRUE) ### Disp = 0.00361 , BCV = 0.0601
de_exact <- exactTest(d)
top_genes <- topTags(de_exact, n=25, adjust.method="BH", sort.by="PValue")
summary(de <- decideTestsDGE(de_exact), adjust.method="BH", p.value=0.05)
detags <- rownames(d)[as.logical(de)]
plotSmear(de_exact, de.tags=detags)
abline(h=c(-1, 1), col="blue")
write.table(topTags(de_exact,n=1000000)$table, file=[output])
```