Tutorial 10: Differential Expression

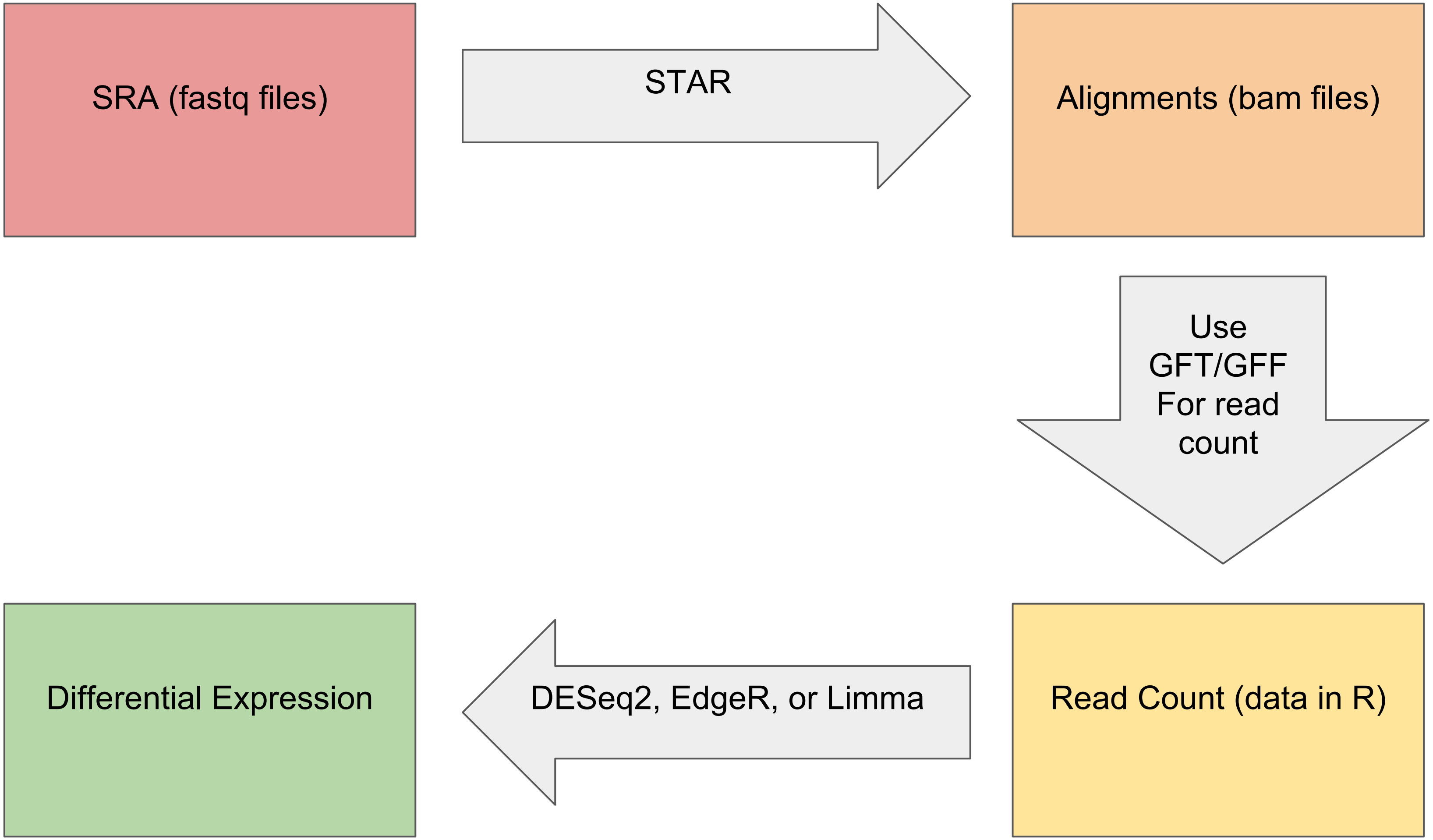
Feiyang Xue
Timothy Merkel

What is Differential Expression?

- A Differential Expression is a way of quantifying and comparing gene expression between conditions. There are three different Differential Expression Methods we will look at: DESeq2, EdgeR, and Limma

- Quantifying and comparing gene expression between conditions is accomplished by analyzing read counts that are created using a variety of different tools: HTSeq, FeatureCounts, Rcount, and more.

Workflow



SRA (fastq files)

STAR

Alignments (bam files)

Use GFT/GFF For read count

Read Count (data in R)

DESeq2, EdgeR, or Limma

Differential Expression

DESeq2: Differential gene expression analysis based on the negative binomial distribution

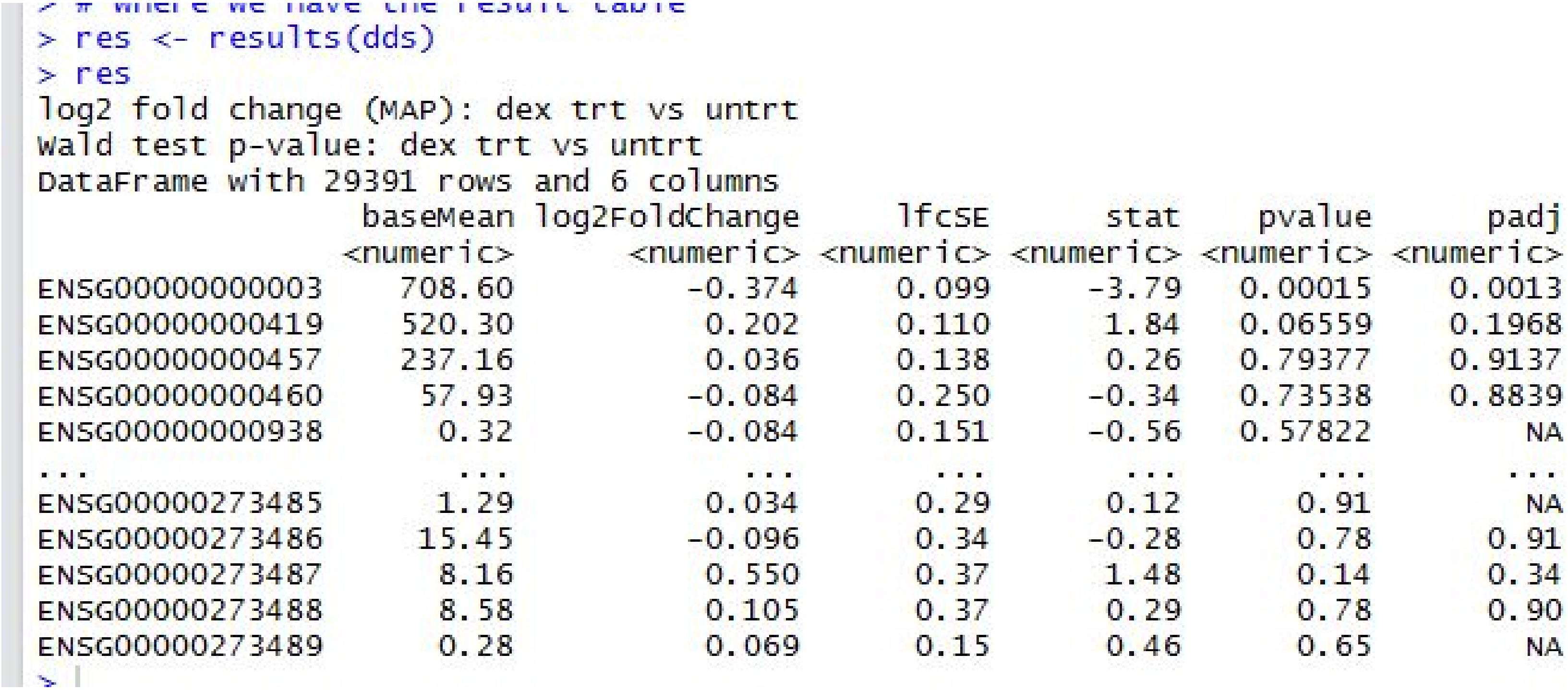
Code taken from https://www.bioconductor.org/help/workflows/rnaseqGene/

Data taken from airway package of R:
https://bioconductor.org/packages/release/data/experiment/html/airway.html

Key steps:

- Prepare input data in BAM format. (samtools -bS)
- Load data with method “summarizeOverlaps” from “GenomicAlignments” package
- Call “DESeqDataSet”, “DESeq”, “results” from “DESeq2” package

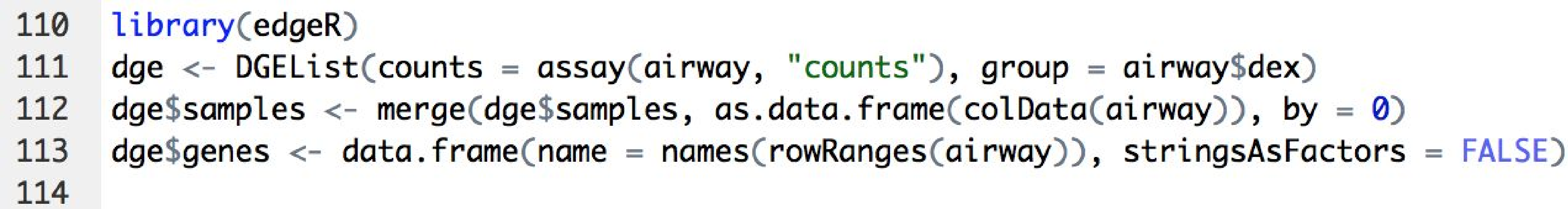
DESeq2 Sample Output



```
> # where we have the result table
> res <- results(dds)
> res
log2 fold change (MAP): dex trt vs untrt
Wald test p-value: dex trt vs untrt
DataFrame with 29391 rows and 6 columns
                 baseMean log2FoldChange     lfcSE      stat    pvalue      padj
                <numeric>      <numeric> <numeric> <numeric> <numeric> <numeric>
ENSG00000000003    708.60         -0.374     0.099     -3.79   0.00015    0.0013
ENSG00000000419    520.30          0.202     0.110      1.84   0.06559    0.1968
ENSG00000000457    237.16          0.036     0.138      0.26   0.79377    0.9137
ENSG00000000460     57.93         -0.084     0.250     -0.34   0.73538    0.8839
ENSG00000000938      0.32         -0.084     0.151     -0.56   0.57822        NA
...                   ...            ...       ...       ...       ...       ...
ENSG00000273485      1.29          0.034      0.29      0.12      0.91        NA
ENSG00000273486     15.45         -0.096      0.34     -0.28      0.78      0.91
ENSG00000273487      8.16          0.550      0.37      1.48      0.14      0.34
ENSG00000273488      8.58          0.105      0.37      0.29      0.78      0.90
ENSG00000273489      0.28          0.069      0.15      0.46      0.65        NA
> 
```

EdgeR

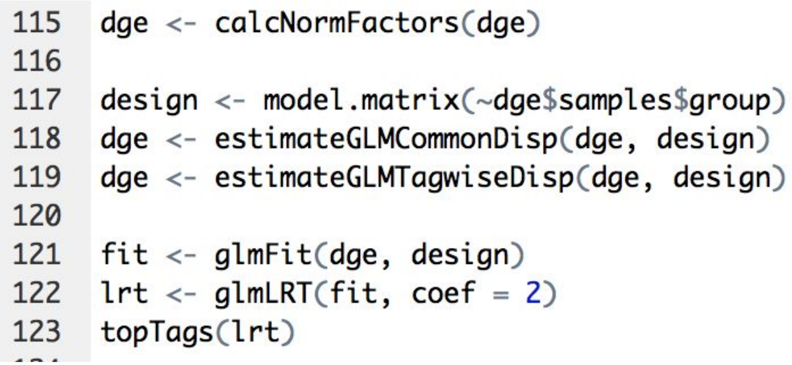
Although EdgeR does not take the *SummarizedExperiment* object that we used for DESeq2 as an input, there is some simple r code that will convert this object to a format that EdgeR can deal with:



```r
library(edgeR)
dge <- DGEList(counts = assay(airway, "counts"), group = airway$dex)
dge$samples <- merge(dge$samples, as.data.frame(colData(airway)), by = 0)
dge$genes <- data.frame(name = names(rowRanges(airway)), stringsAsFactors = FALSE)

```

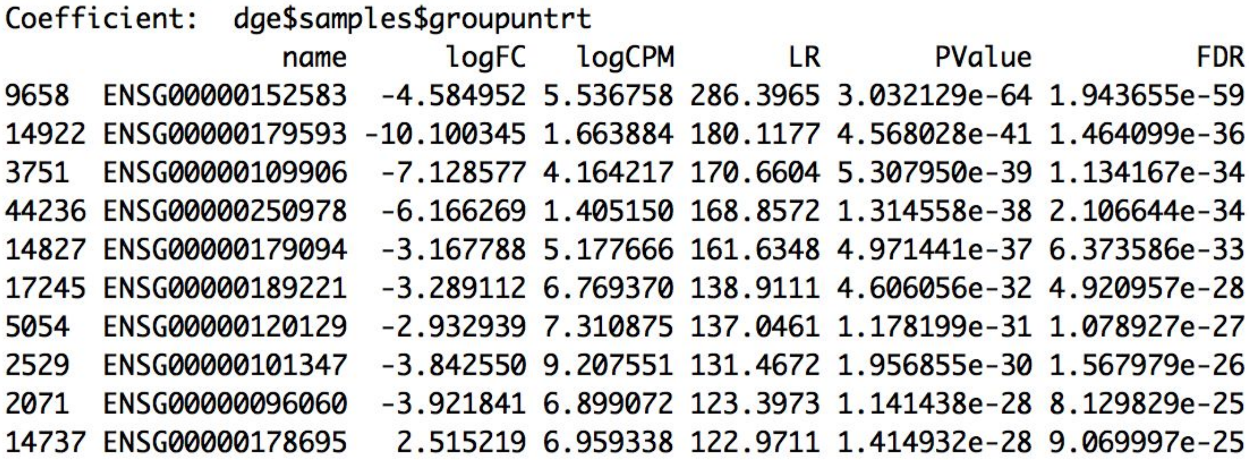
EdgeR

Once the data was in the format that EdgeR can deal with, we ran the Differential Expression code:



```
dge <- calcNormFactors(dge)

design <- model.matrix(~dge$samples$group)
dge <- estimateGLMCommonDisp(dge, design)
dge <- estimateGLMTagwiseDisp(dge, design)

fit <- glmFit(dge, design)
lrt <- glmLRT(fit, coef = 2)
topTags(lrt)
```

Which gives the output:



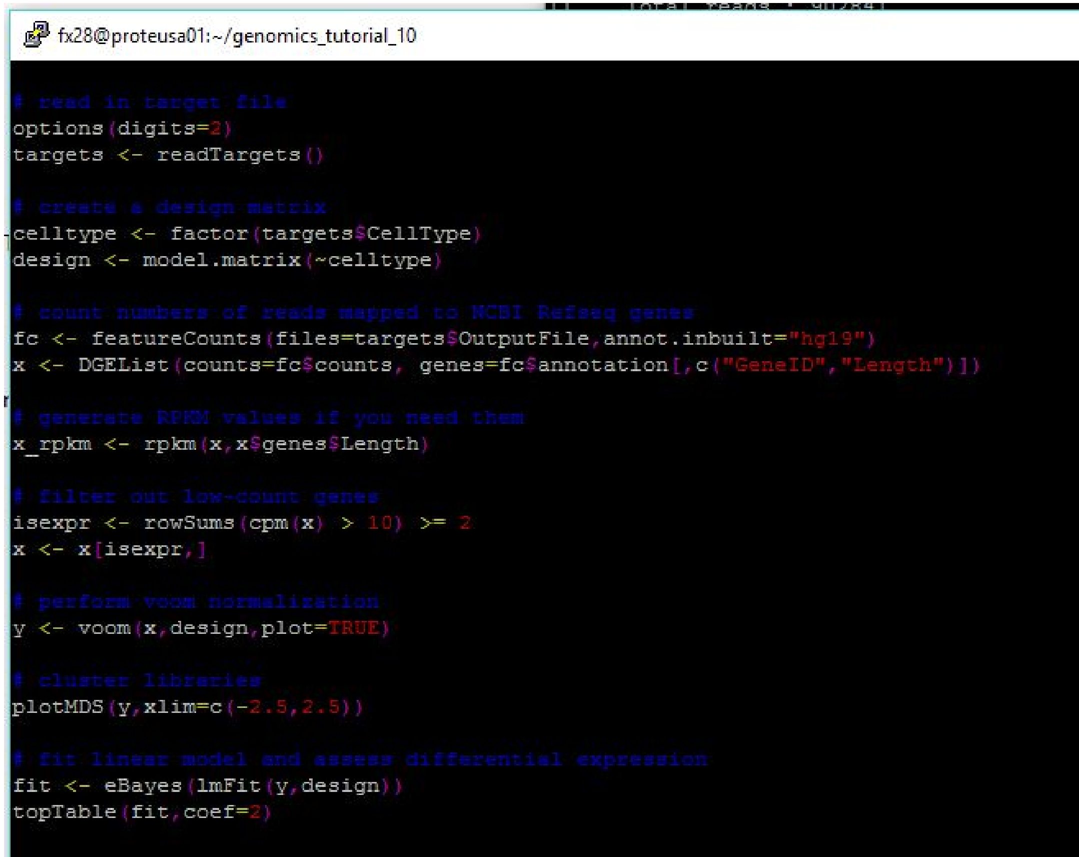
```
Coefficient:  dge$samples$groupuntrt
                 name      logFC   logCPM       LR       PValue          FDR
9658  ENSG00000152583  -4.584952 5.536758 286.3965 3.032129e-64 1.943655e-59
14922 ENSG00000179593 -10.100345 1.663884 180.1177 4.568028e-41 1.464099e-36
3751  ENSG00000109906  -7.128577 4.164217 170.6604 5.307950e-39 1.134167e-34
44236 ENSG00000250978  -6.166269 1.405150 168.8572 1.314558e-38 2.106644e-34
14827 ENSG00000179094  -3.167788 5.177666 161.6348 4.971441e-37 6.373586e-33
17245 ENSG00000189221  -3.289112 6.769370 138.9111 4.606056e-32 4.920957e-28
5054  ENSG00000120129  -2.932939 7.310875 137.0461 1.178199e-31 1.078927e-27
2529  ENSG00000101347  -3.842550 9.207551 131.4672 1.956855e-30 1.567979e-26
2071  ENSG00000096060  -3.921841 6.899072 123.3973 1.141438e-28 8.129829e-25
14737 ENSG00000178695   2.515219 6.959338 122.9711 1.414932e-28 9.069997e-25
```

Limma: Linear Models for Microarray and RNA-Seq Data

Sample code and data taken from: http://bioinf.wehi.edu.au/RNAseqCaseStudy/

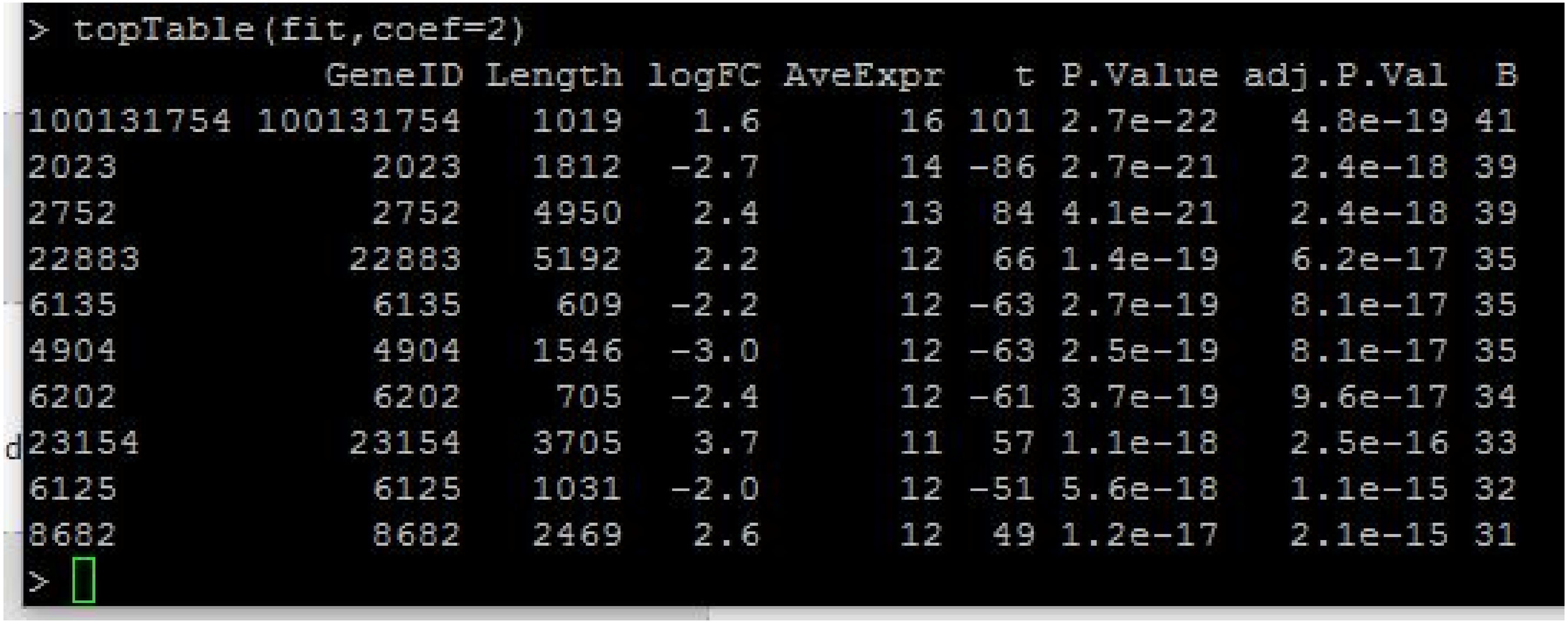
Sample data is read data and reference sequence of human chromosome 1 (GRCh37/hg19)

- Prepare aligned reads as input
- Sample code afterwards:
  - “$OutputFile” is input here
  - In *.bam format
  - “CellType” information is needed



```
# read in target file
options(digits=2)
targets <- readTargets()

# create a design matrix
celltype <- factor(targets$CellType)
design <- model.matrix(~celltype)

# count numbers of reads mapped to NCBI Refseq genes
fc <- featureCounts(files=targets$OutputFile,annot.inbuilt="hg19")
x <- DGEList(counts=fc$counts, genes=fc$annotation[,c("GeneID","Length")])

# generate RPKM values if you need them
x_rpkm <- rpkm(x,x$genes$Length)

# filter out low-count genes
isexpr <- rowSums(cpm(x) > 10) >= 2
x <- x[isexpr,]

# perform voom normalization
y <- voom(x,design,plot=TRUE)

# cluster libraries
plotMDS(y,xlim=c(-2.5,2.5))

# fit linear model and assess differential expression
fit <- eBayes(lmFit(y,design))
topTable(fit,coef=2)
```

Limma Sample Output



```
> topTable(fit,coef=2)
                GeneID Length logFC AveExpr   t P.Value adj.P.Val  B
100131754 100131754   1019   1.6      16 101 2.7e-22   4.8e-19 41
2023           2023   1812  -2.7      14 -86 2.7e-21   2.4e-18 39
2752           2752   4950   2.4      13  84 4.1e-21   2.4e-18 39
22883         22883   5192   2.2      12  66 1.4e-19   6.2e-17 35
6135           6135    609  -2.2      12 -63 2.7e-19   8.1e-17 35
4904           4904   1546  -3.0      12 -63 2.5e-19   8.1e-17 35
6202           6202    705  -2.4      12 -61 3.7e-19   9.6e-17 34
23154         23154   3705   3.7      11  57 1.1e-18   2.5e-16 33
6125           6125   1031  -2.0      12 -51 5.6e-18   1.1e-15 32
8682           8682   2469   2.6      12  49 1.2e-17   2.1e-15 31
> 
```

Differential Expression Method Comparison

- DESeq2, EdgeR, and Limma’s voom are fairly similar, but they handle low counts and outliers slightly differently

- As a result of this, it is generally accepted that EdgeR is preferable for small counts but that Limma is often more reliable when the data is very noisy.

- Speed in this case is not particularly an issue, since we have gotten our data into the count format. Getting the data into the count format is what really takes a while, but all three methods have more or less the same preceding pipeline. All three methods run quickly enough that we did not observe much of a difference.