Gene Expression Analysis for Complex Study Designs

Blythe Durbin-Johnson, Ph.D.
March 12, 2015
Outline

• Overview of gene expression

• Statistical models for microarray and RNA-Seq data

• Tutorial on analysis with limma and edgeR packages
Measuring Gene Expression With Microarrays

• RNA reverse transcribed to cDNA

• Fluorescent dyes incorporated into cDNA during reverse transcription

• cDNA hybridized to probes on microarray

• Measure of expression is brightness of probe
Measuring Gene Expression With RNA-Seq

- RNA fragmented, reverse transcribed to cDNA
- cDNA sequenced using NGS
- Measure of expression for a gene is number of fragments with sequences aligning to that gene
Normal Distribution Models for Microarray Data

• Log-transformed microarray data usually analyzed as if normally distributed:

\[ y_{ij} = \mu_{ij} + \varepsilon_{ij} \]

• \( y_{ij} \) is log expression for sample i, gene j
• \( \mu_{ij} \) is mean log expression for sample i, gene j
• \( \varepsilon_{ij} \) is normally distributed error term with variance \( \sigma_j^2 \)
Normal Distribution Models for Microarray Data

• For two-group experiment:
  \[ \mu_{ij} = \beta_{0j}, \text{ if sample } i \text{ is in control group} \]
  \[ \mu_{ij} = \beta_{0j} + \beta_{1j}, \text{ if sample } i \text{ is in treatment group} \]

• \( \beta_{1j} \) is log fold change for tx vs. control for gene j
Normal Models for Microarray Data

• For a more complex design:

\[ \mu_{ij} = x_i^T \beta_j \]

where \( x_i^T \) is the \( i^{\text{th}} \) row of the “design matrix”
(more later)
Normal Distribution Models for Microarray Data

• The variance $\sigma_j^2$ has to be estimated in order to do significance testing

• Sample size is often too small to do this with data from one gene alone:
  – i.e. the residual standard deviation $s_j^2$ is not a good estimate of $\sigma_j^2$
Normal Models for Microarray Data

- Limma uses empirical Bayes methods to estimate $s_j^2$ by “borrowing information” from other genes
- Assume the variances $\sigma_j^2$ (across all genes) come from an inverse gamma distribution
- Use this “prior” inverse gamma distribution and existing data to get “posterior” estimate of $\sigma_j^2$
Normal Models for Microarray Data

• Parameters of prior distribution are estimated from data (which makes this empirical Bayes)
• Resulting posterior estimate of $\sigma_j^2$ is weighted combination of $s_j^2$ and a variance estimate $s_0^2$ from all the genes:

$$s_g^2 = \frac{d_0 s_0^2 + d_g s_g^2}{d_0 + d_g},$$
NB Models For RNA-Seq Data

• RNA-Seq tag count data are often modelled with a negative binomial (NB) distribution
• NB(r, p) is distribution of number of successes before r failures, if P(success) = p
• NB is also gamma mixture of Poisson distributions
  – Common model for overdispersed count data
  – “Overdispersed” means variance > mean
NB Models for RNA-Seq Data

• $E[y_{ij}] = \mu_{ij}$
Where $y_{ij}$ is the count for sample $i$, gene $j$

• $\text{Var}(y_{ij}) = \mu_{ij} + \phi_j \mu^2_{ij}$

• $\phi_j$ is called the *dispersion parameter*
NB Models for RNA-Seq Data

• For two-group experiment:
  \[
  \log(\mu_{ij}) = \beta_{0j}, \text{ if sample } i \text{ is in control group}
  \]
  \[
  \log(\mu_{ij}) = \beta_{0j} + \beta_{1j}, \text{ if sample } i \text{ is in treatment group}
  \]

• \( \beta_{1j} \) is log fold change for tx vs. control for gene j
NB Models for RNA-Seq Data

• For a more complex design:
  \[ \log(\mu_{ij}) = \mathbf{x}_i^T \mathbf{\beta}_j \]

where \( \mathbf{x}_i^T \) is the \( i^{\text{th}} \) row of the “design matrix”
(more later)
NB Models for RNA-Seq Data

• The dispersion $\phi_j$ has to be estimated in order to do significance testing

• Sample size is often too small to do this with data from one gene alone
NB Models for RNA-Seq Data

• edgeR uses empirical-Bayes-like methods to “borrow information” when estimating $\phi_j$

• The math is very complicated

• Many layers of approximation due to intractability of NB distribution
  – Problematic?
The Design Matrix

• Recall that for microarrays:
  – the mean log expression for sample i, gene j can be stated as $x_i^T \beta_j$

• For RNA-Seq data:
  – the log mean count for sample i, gene j can be stated as $x_i^T \beta_j$

Where $x_i^T$ is the $i^{th}$ row of the “design matrix”
The Design Matrix

- What is the design matrix?
- It has one row for each sample
- Columns have information about experimental design, phenotype, etc.
- Design matrix is assumed to be the same for each gene
- Using limma and edgeR requires some understanding of the design matrix
The Design Matrix

For a two-group study with 3 replicates per group:

\[
\begin{align*}
\mu_{ij} &= \beta_{0j}, \text{ if sample } i \text{ is in control group} \\
\mu_{ij} &= \beta_{0j} + \beta_{1j}, \text{ if sample } i \text{ is in treatment group}
\end{align*}
\]

Design matrix =

\[
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 1 \\
1 & 1 \\
1 & 1
\end{bmatrix}
\]

\[\beta_0 \quad \beta_1\]
The Design Matrix

- What if we want to model expression by diagnosis, gender, and age?

\[ \mathbf{x}_i^T \beta_j = \beta_{0j} + \beta_{1j} \cdot I(dx = TD) + \beta_{2j} \cdot I(gender = M) + \beta_{3j} \cdot \text{age} \]
The Design Matrix

– Subject 1: ASD, male, 37 months
– Subject 2: TD, female, 40 months
– Subject 3: ASD, female, 48 months

• First three rows of design matrix would be:

\[
\begin{bmatrix}
1 & 0 & 1 & 37 \\
1 & 1 & 0 & 40 \\
1 & 0 & 1 & 48 \\
\end{bmatrix}
\]

Subject 1

Subject 2

Subject 3
Normal Models for RNA-Seq Data

• Limma’s `voom()` function calculates variance weights for log counts

• Weights estimated from fitted mean-variance trend + covariate information

• Allows RNA-Seq data to be used in limma
voom: Mean-variance trend
Normal Models for RNA-Seq Data

- Limma-voom performs well compared to packages based on NB distribution
- Normal approximation may not work as well for very low counts
- Filtering of very low counts typically recommended before using limma-voom
A Note on Multiple Testing

• Suppose we conduct tests comparing expression between 2 groups, for 24K genes

• Would expect $0.05 \times 24,000 = 1200$ genes to have $P < 0.05$ under null conditions

• $P < 0.05$ not a useful definition of statistical significance in this setting
A Note on Multiple Testing

• Bonferroni correction would define significance as $P < 0.05/24000$

• Too conservative!
A Note on Multiple Testing

• In ‘omics studies, significance often defined in terms of false discovery rate:
  – E.g. adjust p-values such that expected proportion of false discoveries among a list of genes with adjusted-\(p\) \(< 0.05 = 5\%\)

• How much a p-value for a given gene is adjusted depends on its rank in the list, size of other, smaller p-values

• Look at the FDR adjusted p-values, not the raw p-values
Microarray Preprocessing

• Microarray preprocessing is platform specific
• Most easily done in software provided by platform developer:
  – Illumina’s GenomeStudio
  – Affymetrix’s GeneChip Command Console
• RMA is generally good choice for normalization method
Microarray Preprocessing

- Output normalized data as text file for input to R
- File will have gene/probe name as first column, then one column per sample
- One row per gene/probe
RNASeq Preprocessing

• Not for the uninitiated
• Huge files, long computing times
• Genome Center Bioinformatics Core gives courses on analysis in Galaxy (GUI-based platform)

OR

• Hire them (or someone else) to preprocess data for you
  – A few billable hours vs. months of your time
RNASeq Preprocessing

• Output from sequencer are .fastq files
  – Multiple files per sample
  – Contain sequences of each cDNA fragment + quality information
  – 1—3 GB in size

• QC, adapter trimming

• Align to genome or transcriptome
RNASeq Preprocessing

• Output of alignment is .bam file
  – One file per sample
  – About 2 GB per file

• Use HTSeq or other program to count number of fragments that align to a given gene

• Output is text file of counts with one row per gene/transcript, one column per sample
RNASeq Preprocessing

- WARNING: Output from Cufflinks cannot be used in limma-voom or edgeR
- Check that your counts table has integer counts (no decimals)
Setup

• Install R following instructions here: http://cran.r-project.org/
• Current version is 3.1.3
• Updated several times a year
• Know what version you are using, state this in manuscripts
  – This is extra important for contributed packages like limma and edgeR
Setup

• Install Bioconductor by starting R and entering the commands:
  > source("http://bioconductor.org/biocLite.R")
  > biocLite()

• Install limma and edgeR:
  > biocLite("limma")
  > biocLite("edgeR")
Microarray Analysis With Limma

• Load limma:
  > library(limma)

• Read your normalized gene expression data into R:
  > dat <- read.delim("mydata.txt")
# Microarray Analysis With Limma

```r
> head(dat)

<table>
<thead>
<tr>
<th></th>
<th>X5645570027_A.AVG_Signal</th>
<th>X5645570027_B.AVG_Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILMN_1762337</td>
<td>7.145766</td>
<td>7.081886</td>
</tr>
<tr>
<td>ILMN_2055271</td>
<td>7.076804</td>
<td>7.045860</td>
</tr>
<tr>
<td>ILMN_1736007</td>
<td>7.006162</td>
<td>6.962973</td>
</tr>
<tr>
<td>ILMN_2383229</td>
<td>7.097061</td>
<td>7.032479</td>
</tr>
<tr>
<td>ILMN_1806310</td>
<td>6.967002</td>
<td>7.169537</td>
</tr>
<tr>
<td>ILMN_1779670</td>
<td>6.840718</td>
<td>6.897797</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X5645570027_C.AVG_Signal</th>
<th>X5645570027_D.AVG_Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILMN_1762337</td>
<td>7.029071</td>
<td>7.247738</td>
</tr>
<tr>
<td>ILMN_2055271</td>
<td>7.126903</td>
<td>7.065131</td>
</tr>
<tr>
<td>ILMN_1736007</td>
<td>7.075009</td>
<td>7.002421</td>
</tr>
<tr>
<td>ILMN_2383229</td>
<td>7.084756</td>
<td>7.280248</td>
</tr>
<tr>
<td>ILMN_1806310</td>
<td>7.052176</td>
<td>6.947051</td>
</tr>
<tr>
<td>ILMN_1779670</td>
<td>6.947104</td>
<td>6.949196</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X5645570027_E.AVG_Signal</th>
<th>X5645570027_F.AVG_Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILMN_1762337</td>
<td>7.075025</td>
<td>7.140636</td>
</tr>
</tbody>
</table>
```
Microarray Analysis With Limma

• Read your phenotype/clinical information into R (we’ll assume it’s in a .xlsx file):

```r
> install.packages("gdata")
# If not installed yet yet
> library(gdata)
> clindat <-
read.xls("clinicaldata.xlsx",
        stringsAsFactors = F)
```
Microarray Analysis With Limma

> head(clindat)
  X subj_group_id dx gender ChildRace MmRace DadRace ChildAgeAtConsent ChildAgeAtAssesment ChildAgeAtBloodDraw ChildAgeAtIntStart
  1 1 1 1 M 1 1 4 4 1532 1574 1574 1594.000000000000000000
  2 1 1 1 M 1 1 6 6 1209 1237 1237 1293.000000000000000000
  3 1 1 1 M 1 1 1 1 1050 1123 1123 1135.000000000000000000
  4 1 1 1 M 1 1 6 6 1412 1615 1615 1522.000000000000000000
  5 1 1 1 M 1 1 1 1 1715 1757 1757 1802.000000000000000000
  6 1 1 1 M 9 9 9 9 1659 1818 1818 1986.000000000000000000

  ChildAgeAtIntEnd AgeMm AgeDad YOB MmMmD Edu DadEdu MaxEdu PaymentDelivery BirthPlaceMm OwnHome
  1 1594.000000000000000000 26.78751 28.454794520547946 1999 3 3 3 2 2 1 0.000000000000000
  2 1253.000000000000000000 17.81918 21.775342465753425 1999 2 1 2 2 2 1 1.000000000000000
  3 1135.000000000000000000 34.64556 34.596249719290363 2000 4 5 5 5 5 2 1 1.000000000000000
  4 1547.000000000000000000 35.84658 37.717808219178082 1989 5 4 5 5 5 2 3 1.000000000000000
  5 1802.000000000000000000 41.27397 43.726027397260275 1999 3 3 3 3 3 2 1 1.000000000000000
  6 1967.000000000000000000 36.77260 41.788344935998204 1998 5 4 5 5 5 2 1 1.000000000000000

  Total_Children_Born locale birth_regcr RCgroup RCprimary EnrollmentRC EnrollmentRC_source EnrollmentRCgroup GA_Day Rev
  1 2 1 360 3 North Bay 371 Specified 2 280.000000000000000000
  2 1 1 371 2 North Bay 371 Specified 2 283.000000000000000000
  3 1 1 371 2 North Bay 371 Specified 2 256.000000000000000000
  4 2 2 360 5 Eastern LA 373 Specified 5 271.000000000000000000
  5 3 2 368 5 Eastern LA 373 Specified 5 284.000000000000000000
  6 1 2 360 5 Eastern LA 373 Specified 5 276.000000000000000000

  GA wk rev Dx
  1 40.000000000000000000 ASD
  2 40.480000000000000000 ASD
  3 38.570000000000000000 ASD
  4 38.710000000000000001 ASD
  5 40.570000000000000000 ASD
  6 39.480000000000000000 ASD
Microarray Analysis With Limma

• Phenotype data MUST:
  1. Have one row for each column of the microarray/RNA-Seq data
  2. Have rows IN THE SAME ORDER as the columns of the microarray/RNA-Seq data
  3. Have no missing values

• (1) and (3) will generate errors, (2) will not. Be careful!
Microarray Analysis With Limma

Set up design matrix, for model with Dx, gender, and child’s age at blood draw:

```r
>design <- model.matrix(~Dx + gender + ChildAgeAtBloodDraw, data = clindat)
```
An Aside on Models in R

• \( \sim \text{var1} + \text{var2} \) denotes a model with main effects for var1 and var2
• \( \sim \text{var1} + \text{var2} + \text{var1:var2} \) denotes a model with main effects for var1 and var2 and their interaction
• \( \sim \text{var1*var2} \) is the same as
  \( \sim \text{var1} + \text{var2} + \text{var1:var2} \)
• * notation has different meaning than in some other software
An Aside on Models in R

• For categorical variables, category that is first alphabetically is treated as reference group unless otherwise specified
  – Different from SAS

• Can change this by reordering factor levels:

```r
> clindat$Dx <- factor(clindat$Dx,
                        levels = c("TD", "ASD"))

  – This makes TD the reference group
Microarray Analysis With Limma

> head(design)

(Intercept)  DxD TD genderM  ChildAgeAtBloodDraw
1       1  0  1 1574
2       1  0  1 1237
3       1  0  1 1123
4       1  0  1 1433
5       1  0  1 2103
6       1  0  1 1944

• Design matrix has one row per sample/subject
• Everyone has 1 in 1st column
• TD subjects have 1 in second column
• Male subjects have 1 in 3rd column
# Microarray Analysis With Limma

- **Fit model**

  ```r
  fit <- lmFit(dat, design)
  fit
  ```

  An object of class "MArrayLM"

  ```r
  $coefficients
  (Intercept)  DxTD  genderM  ChildAgeAtBloodDraw
  ILMN_1762337 3.2437811 -1.0178656 -1.2542614  0.001220603
  ILMN_2055271 13.7515253 -0.4530049 -0.4320275 -0.000636068
  ILMN_1736007 -2.5340422  0.8934152  0.8714748  0.001423782
  ILMN_2383229  0.3321417 -1.0065782  0.3172549  0.002065563
  ILMN_1806310  8.1339898 -1.6708637 -3.2468605 -0.001543164
  47226 more rows ...
  ```

  $rank
  ```r
  [1] 4
  ```
Microarray Analysis With Limma

- Calculate empirical Bayes variance estimates:
  > fit <- eBayes(fit)

- Test for effect of diagnosis, adjusting for gender and age:
  > table <- topTable(fit, coef=2)
Microarray Analysis With Limma

```r
> table

<table>
<thead>
<tr>
<th></th>
<th>logFC</th>
<th>AveExpr</th>
<th>t</th>
<th>P.Value</th>
<th>adj.P.Val</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILMN_1696302</td>
<td>141.022586</td>
<td>204.647010</td>
<td>4.916709</td>
<td>1.645558e-06</td>
<td>0.05963238</td>
<td>4.350389</td>
</tr>
<tr>
<td>ILMN_3212051</td>
<td>9.979139</td>
<td>17.014737</td>
<td>4.823597</td>
<td>2.525137e-06</td>
<td>0.05963238</td>
<td>3.996236</td>
</tr>
<tr>
<td>ILMN_2141157</td>
<td>7.469358</td>
<td>7.671294</td>
<td>4.724316</td>
<td>3.959434e-06</td>
<td>0.06233601</td>
<td>3.624566</td>
</tr>
</tbody>
</table>
```

- logFC = logFC for TD vs. ASD
- AveExpr = average expression across all samples
- t = moderated t statistic (using empirical Bayes estimates for s.d. and degrees of freedom)
- P.Value = UNADJUSTED P.Value
- adj.P.Val = FDR adjusted P-value <- this is the one to look at!
- B = estimated log odds that gene is DE (take with a grain of salt)
Microarray Analysis With Limma

• Write file with results for all genes, sorted by p-value:

```r
> table.all <- topTable(fit, 
   coef = 2, n = Inf)
> write.table(table.all, 
   file = "outfile.txt", 
   sep = "\t", quote = F)
```
RNA-Seq Analysis With Limma-Voom

• load edgeR (which loads limma):
  >library(edgeR)

• Read in counts table:
  >read.delim("mydata.txt")
RNA-Seq Analysis With Limma-Voom

```r
> head(dat)

<table>
<thead>
<tr>
<th>CHG0836.aln.sorted.bam.counts</th>
<th>Run20.CHG0479.aln.sorted.bam.counts</th>
<th>Run20.CHG0495.aln.sorted.bam.counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1BG</td>
<td>80</td>
<td>66</td>
</tr>
<tr>
<td>A1BG-AS1</td>
<td>38</td>
<td>22</td>
</tr>
<tr>
<td>A1CF</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>A2M</td>
<td>32</td>
<td>13</td>
</tr>
<tr>
<td>A2M-AS1</td>
<td>24</td>
<td>10</td>
</tr>
<tr>
<td>A2ML1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A1BG</td>
<td>99</td>
<td>109</td>
</tr>
<tr>
<td>A1BG-AS1</td>
<td>57</td>
<td>45</td>
</tr>
<tr>
<td>A1CF</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>A2M</td>
<td>16</td>
<td>42</td>
</tr>
<tr>
<td>A2M-AS1</td>
<td>19</td>
<td>41</td>
</tr>
<tr>
<td>A2ML1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Run20.CHG0568.aln.sorted.bam.counts</td>
<td>Run20.CHG0586.aln.sorted.bam.counts</td>
<td>Run20.CHG0617.aln.sorted.bam.counts</td>
</tr>
<tr>
<td>A1BG</td>
<td>62</td>
<td>108</td>
</tr>
<tr>
<td>A1BG-AS1</td>
<td>32</td>
<td>41</td>
</tr>
<tr>
<td>A1CF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A2M</td>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>A2M-AS1</td>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>A2ML1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Run20.CHG0628.aln.sorted.bam.counts</td>
<td>Run20.CHG0649.aln.sorted.bam.counts</td>
<td>Run20.CHG0651.aln.sorted.bam.counts</td>
</tr>
<tr>
<td>A1BG</td>
<td>83</td>
<td>94</td>
</tr>
<tr>
<td>A1BG-AS1</td>
<td>38</td>
<td>44</td>
</tr>
<tr>
<td>A1CF</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>A2M</td>
<td>9</td>
<td>20</td>
</tr>
<tr>
<td>A2M-AS1</td>
<td>11</td>
<td>20</td>
</tr>
<tr>
<td>A2ML1</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>
```
RNA-Seq Analysis With Limma-Voom

• Read in phenotype data:
  ```
  >library(gdata)
  >clindat <- read.xls("clinicaldata.xlsx", stringsAsFactors = F)
  ```

• Create design matrix
  ```
  >design <- model.matrix(~Dx + gender + ChildAgeAtBloodDraw, data = clindat)
  ```
RNA-Seq Analysis With Limma-Voom

• Filter out genes with average count < 1:
  > lowcounts <- which(rowSums(dat)/ncol(dat)< 1)
  > dat.filt <- dat[-lowcounts,]

• Convert count matrix to a DGE List object:
  > DGE <- DGEList(dat.filt)

• Calculate normalization factors:
  > DGE <- calcNormFactors(DGE)
RNA-Seq Analysis With Limma-Voom

• Use voom function to calculate variance weights:

\[ v \leftarrow \text{voom}(\text{DGE}, \text{design}, \text{plot} = \text{T}) \]
voom: Mean-variance trend
voom: Mean-variance trend

This is why you need to filter low counts
RNA-Seq Analysis With Limma-Voom

• Pass output from voom into limma’s lmFit function:

```r
> fit <- lmFit(v, design)
```  

• Proceed as for a microarray analysis
RNA-Seq Analysis With edgeR

• First few steps are similar to limma-voom:

```r
library(edgeR)
dat <- read.delim("mydata.txt")
library(gdata)
clindat <- read.xls("clinicaldata.xlsx", stringsAsFactors = F)

design <- model.matrix(~Dx + gender + ChildAgeAtBloodDraw, data = clindat)
```
RNA-Seq Analysis With edgeR

• Filter observations with very low counts (maybe less crucial here):

  > lowcounts <- which(rowSums(dat)/ncol(dat)< 1)
  > dat.filt <- dat[-lowcounts,]

• Convert count matrix to a DGE List object:

  > DGE <- DGEList(dat.filt)

• Calculate normalization factors:

  > DGE <- calcNormFactors(DGE)
RNA-Seq Analysis With edgeR

• Calculate smoothed dispersion parameters:
  > DGE <- estimateDisp(DGE, design)

• Fit negative binomial GLM
  > fit <- glmFit(DGE, design)
RNA-Seq Analysis With edgeR

- Likelihood ratio test for differential expression between diagnoses
  
  ```r
  fit2 <- glmLRT(fit, coef = 2)
  ```

- Show most significantly DE genes
  
  ```r
  table <- topTags(fit2)
  ```
RNA-Seq Analysis With edgeR

```r
> table
Coefficient: DxTD

<table>
<thead>
<tr>
<th>Gene</th>
<th>logFC</th>
<th>logCPM</th>
<th>LR</th>
<th>PValue</th>
<th>FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD177</td>
<td>1.6509785</td>
<td>3.87230914</td>
<td>23.75014</td>
<td>1.096876e-06</td>
<td>0.01860959</td>
</tr>
<tr>
<td>LGALS4</td>
<td>0.5679173</td>
<td>0.32851413</td>
<td>19.10747</td>
<td>1.235602e-05</td>
<td>0.09650873</td>
</tr>
<tr>
<td>CLRN1-AS1</td>
<td>1.4291805</td>
<td>-2.74928526</td>
<td>17.94957</td>
<td>2.268361e-05</td>
<td>0.09650873</td>
</tr>
</tbody>
</table>
```

- **logFC** = log fold change for TD vs. ASD
- **logCPM** = average log counts per million over all samples
- **LR** = likelihood ratio
- **P-value** = unadjusted p-value from LRT
- **FDR** = FDR-adjusted p-value from LRT
RNA-Seq Analysis With edgeR

• To write output for all genes to a file:

```r
> table.all <- topTags(fit2, n=Inf)
> write.table(table.all,
    file = "output.txt",
    sep = "\t", quote = F)
```
Now What?

- Next steps often involve enrichment analyses/pathway analysis
- GSEA (free):
  http://www.broadinstitute.org/gsea/index.jsp
  – Use GSEAPreranked for complex study designs
- topGO R package (free):
Now What

- GOStats R package (free)

- Ingenuity Pathway Analysis (not free)

- Many other free and commercial resources
Other Resources

• Limma vignette

• edgeR vignette

• Davis R Users Group:
  http://www.noamross.net/davis-r-users-group.html