Bulk RNA-seq analysis

Charlotte Soneson

Friedrich Miescher Institute for Biomedical Research &
SIB Swiss Institute of Bioinformatics
Differential analysis types for RNA-seq

- Does the total output of a gene change between conditions? **DGE**
- Does the expression of individual transcripts change? **DTE**
- Does any isoform of a given gene change? **DTE+G**
- Does the isoform composition for a given gene change? **DTU/DIU/DEU**
- (Does anything change? **GDE***)

- need **different** abundance quantification of transcriptomic features (genes, transcripts, exons)

*https://liorpachter.wordpress.com/2018/02/15/gde%C2%B2-dge%C2%B2-dtu%C2%B2-dte%E2%82%81%C2%B2-dte%E2%82%82%C2%B2/*
Differential analysis types for RNA-seq

**DGE**
(also DTE)

**DTE**
(also DGE, DTU)

**DTU**
(also DTE)

---

expression

cond 1  cond 2

expression

cond 1  cond 2

expression

cond 1  cond 2

isoform A  isoform B
Differential expression analysis

- Input: expression/abundance matrix (features x samples) + grouping/sample annotation

<table>
<thead>
<tr>
<th>Feature</th>
<th>SRR1039508</th>
<th>SRR1039509</th>
<th>SRR1039512</th>
<th>SRR1039513</th>
<th>SRR1039516</th>
<th>SRR1039517</th>
<th>SRR1039520</th>
<th>SRR1039521</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENSG00000000003</td>
<td>693</td>
<td>451</td>
<td>887</td>
<td>416</td>
<td>1148</td>
<td>1069</td>
<td>774</td>
<td>581</td>
</tr>
<tr>
<td>ENSG00000000005</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ENSG00000000419</td>
<td>466</td>
<td>515</td>
<td>623</td>
<td>364</td>
<td>590</td>
<td>794</td>
<td>419</td>
<td>510</td>
</tr>
<tr>
<td>ENSG00000000457</td>
<td>326</td>
<td>274</td>
<td>372</td>
<td>223</td>
<td>356</td>
<td>450</td>
<td>308</td>
<td>297</td>
</tr>
<tr>
<td>ENSG00000000460</td>
<td>91</td>
<td>75</td>
<td>61</td>
<td>48</td>
<td>110</td>
<td>95</td>
<td>100</td>
<td>82</td>
</tr>
<tr>
<td>ENSG00000000938</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Output: result table (one line per feature)

<table>
<thead>
<tr>
<th>Feature</th>
<th>logFC</th>
<th>logCPM</th>
<th>LR</th>
<th>PValue</th>
<th>FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENSG000000109006</td>
<td>-5.882117</td>
<td>4.12649</td>
<td>924.1622</td>
<td>5.486794e-203</td>
<td>3.493826e-198</td>
</tr>
<tr>
<td>ENSG000000165995</td>
<td>-3.236681</td>
<td>4.603028</td>
<td>576.1025</td>
<td>2.641667e-127</td>
<td>8.410672e-123</td>
</tr>
<tr>
<td>ENSG000000189221</td>
<td>-3.316900</td>
<td>6.718559</td>
<td>562.9504</td>
<td>1.909251e-124</td>
<td>4.052512e-120</td>
</tr>
<tr>
<td>ENSG000000101347</td>
<td>-3.759992</td>
<td>9.290645</td>
<td>449.9697</td>
<td>7.323472e-100</td>
<td>7.772231e-96</td>
</tr>
<tr>
<td>ENSG000000211445</td>
<td>-3.755669</td>
<td>9.102440</td>
<td>433.4656</td>
<td>2.861624e-96</td>
<td>2.603318e-92</td>
</tr>
<tr>
<td>ENSG000000162692</td>
<td>3.616656</td>
<td>4.551120</td>
<td>462.0266</td>
<td>1.994189e-89</td>
<td>1.587360e-85</td>
</tr>
<tr>
<td>ENSG000000171819</td>
<td>-5.705289</td>
<td>3.474627</td>
<td>389.3431</td>
<td>1.150502e-86</td>
<td>8.140055e-83</td>
</tr>
<tr>
<td>ENSG000000152583</td>
<td>-4.364255</td>
<td>5.409103</td>
<td>376.1995</td>
<td>8.363745e-84</td>
<td>5.325782e-80</td>
</tr>
</tbody>
</table>
Most RNA-seq methods (e.g., edgeR, DESeq2, voom) need **raw counts** (or equivalent) as input.

Don’t provide these methods with (e.g.) RPKMs, FPKMs, TPMs, CPMs, log-transformed counts, normalized counts, …

Read documentation carefully!
Why not only relative abundances?

- Ex: ratio between two Poisson distributed variables
  - Low: mean = 20 vs mean = 10
  - High: mean = 2000 vs mean = 1000
Challenges for RNA-seq data analysis

• Choice of statistical distribution
• Normalization between samples
• Few samples -> difficult to estimate parameters (e.g., variance)
• High dimensionality (many genes) -> many tests
Challenges for RNA-seq data analysis

- **Choice of statistical distribution**
- Normalization between samples
- Few samples -> difficult to estimate parameters (e.g., variance)
- High dimensionality (many genes) -> many tests
Characteristics of RNA-seq data

• Variance depends on the mean count

• Counts are non-negative and often highly skewed
Modeling counts - the Poisson distribution

\[ P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \]
Modeling counts - the Poisson distribution

- A famous use of the Poisson distribution was given by von Bortkiewicz (1898) in *Das Gesetz der kleiner Zahlen*

- He studied the number of soldiers in the Prussian army who got kicked by horses, over a number of years and corps

<table>
<thead>
<tr>
<th># horsekicks (k)</th>
<th># obs</th>
<th>fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>109</td>
<td>0.545</td>
</tr>
<tr>
<td>1</td>
<td>65</td>
<td>0.325</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>0.11</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.015</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.005</td>
</tr>
</tbody>
</table>
Modeling counts - the Poisson distribution

• A famous use of the Poisson distribution was given by von Bortkiewicz (1898) in *Das Gesetz der kleiner Zahlen*

• He studied the number of soldiers in the Prussian army who got kicked by horses, over a number of years and corps

<table>
<thead>
<tr>
<th># horsekicks (k)</th>
<th># obs</th>
<th>fraction</th>
<th>$\frac{0.61^k}{k!} e^{-0.61}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>109</td>
<td>0.545</td>
<td>0.543</td>
</tr>
<tr>
<td>1</td>
<td>65</td>
<td>0.325</td>
<td>0.331</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>0.11</td>
<td>0.101</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.015</td>
<td>0.0206</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.005</td>
<td>0.00313</td>
</tr>
</tbody>
</table>
For RNA-seq data:

- reads ~ soldiers
- mapping to gene A ~ being kicked by a horse

Assumes that the probability of a read mapping to gene A is the same for all samples within a class.
Modeling counts

- **Poisson distribution**
  - Quantifies sampling variability
  - \( \text{var}(X) = \mu \)
  - Represents technical replicates well (mRNA proportions are identical across samples)

Example from SEQC data, same sample sequenced across multiple lanes
Modeling counts - the Negative Binomial distribution

\[ P(X = k) = \binom{k + r - 1}{k} \cdot (1 - p)^r p^k \]

Generalizes the Poisson distribution
Modeling counts

• **Negative binomial distribution**
  • \( \text{var}(X) = \mu + \theta \mu^2 \)
  • \( \theta \) = dispersion
  • \( \sqrt{\theta} = "\text{biological coefficient of variation}" \)
• Allows mRNA proportions to vary across samples
• Captures variability across biological replicates better

Example from SEQC data, replicates of the same RNA mix
With count data…

- *linear* modeling (and thus t-tests, ANOVA, etc) is no longer suitable for inference

- *Generalized linear models* to the rescue!
A crash course on GLMs

• A GLM consists of three parts:

  • A distribution, specifying the conditional distribution of the response Y given the predictor values

  • A linear predictor

    \[ \eta = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \]

  • A link function \( g \), linking the conditional expected value of Y to \( \eta \):

    \[ g(\text{E}[Y|X]) = \eta \]
The linear model is a GLM

- A GLM consists of three parts:
  
  - A *distribution*, specifying the conditional distribution of the response $Y$ given the predictor values (Gaussian)
  
  - A *linear predictor*
    
    $$\eta = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$
  
  - A *link function* $g$, linking the conditional expected value of $Y$ to $\eta$: $g(E[Y|X]) = \eta$ (Identity function)
Other commonly used GLMs

- Logistic regression - binary response
  - Binomial distribution
  - logit link function
- Loglinear regression - count response
  - Poisson distribution
  - log link function
GLMs for RNA-seq

- Negative Binomial distribution
- Log link function
- Implemented e.g. in edgeR and DESeq2
The link function in the GLM transforms the mean, not the observed values.

Thus, we can transform the systematic part without changing the assumptions on the random part.

By transforming the response (the observed values), we change also the random part (e.g., the association between mean and variance).
• Instead of modeling the counts, we can **transform them** to a suitable scale and model them with a normal distribution ("microarray-like").

• **voom** (part of the limma package) calculates logCPM values

\[
y_{gi} = \log_2 \left( \frac{r_{gi} + 0.5}{R_i + 1.0 \times 10^6} \right)
\]

• Transformed data is heteroskedastic (variance depends on mean) - use weighted least squares
voom - mean/variance relationship

voom: Mean–variance trend

\[ \text{log}_2(\text{count size} + 0.5) \]
Data transformations - log

Counts

Log2 transform

Data transformations - log
Data transformations - DESeq2

- Two approaches: rlog, variance stabilizing transformation
- Aim: remove dependence of variance on mean after transformation
Nonparametric tests

- Do no assume a specific distribution for the data
- Significance is inferred via permutations
- Perform well for highly replicated experiments
- Lack power for experiments with few replicates
- Resistant to outliers
Challenges for RNA-seq data analysis

- Choice of statistical distribution
- **Normalization between samples**
- Few samples -> difficult to estimate parameters (e.g., variance)
- High dimensionality (many genes) -> many tests
Normalization

- Observed counts depend on:
  - abundance
  - gene length
  - sequencing depth
  - sequencing biases
  - ...
  - "As-is", not directly comparable across samples
Normalization

\[ C_{ij} \sim NB(\mu_{ij} = s_{ij} q_{ij}, \theta_i) \]

- \( s_{ij} \) is a normalization factor (or offset) in the model
- counts are not explicitly scaled
- important exception: voom/limma (followed by explicit modeling of mean-variance association)
Assume that we have RNA-seq reads for one gene. Is the gene differentially expressed?

```r
count.data <- data.frame(counts = c(369, 287, 348, 433, 555, 294, 419),
                         cond = c("1", "1", "1", "1", "2", "2", "2"))
glm.pois <- glm(counts ~ cond, family = poisson, data = count.data)
coefficients(summary(glm.pois))
```

```
##             Estimate Std. Error  z value Pr(>|z|)   
## (Intercept)   5.8840      0.02638  223.050  0.000e+00
## cond2        0.1626      0.03853   4.219   2.451e-05
```
Simple example - offsets

- Relate counts to library sizes
Incorporate library size as offset

```r
count.data$lib.size <- c(3040296, 2717092, 3016179, 3707895, 4422272, 3467730, 3879114)
glm.pois <- glm(counts ~ cond + offset(log(lib.size)), family = poisson, data = count.data)
coefficients(summary(glm.pois))
```

```
##             Estimate Std. Error  z value  Pr(>|z|)
## (Intercept)   -9.06944   0.02638  -343.802  0.00000
## cond2         -0.06635   0.03853  -1.722   0.08506
```
Why offset rather than scaling?

- Variance/Mean for Poisson distributed variable, before as well as after multiplying the values with 2.
Why offset rather than scaling?

- Poisson distributed variables with different means, scaled to have mean = 100

- Raw count
  - mean = 100
  - mean = 1000, scaled by 0.1
  - mean = 10, scaled by 10

Why offset rather than scaling?
How to calculate normalization factors?

• Attempt 1: **total count** (library size)

• Define a reference sample (one of the observed samples or a “pseudo-sample”) - gives a “target library size”

• Normalization factor for sample $j$ is defined by

$$\frac{\text{total count in sample } j}{\text{total count in reference sample}}$$
The influence of RNA composition

- Observed counts are relative
- High counts for some genes are “compensated” by low counts for other genes
How to calculate normalization factors?

- Attempt 2: total count (library size) * compensation for differences in composition

- Idea: use only non-differentially expressed genes to compute the normalization factor

- Implemented by both edgeR (TMM) and DESeq2 (median count ratio)

- Both these methods assume that most genes are not differentially expressed
How to calculate normalization factors?

- Attempt 2: total count (library size) * compensation for differences in composition
“Normalization factors” (edgeR) vs “size factors” (DESeq2)
“Effective library sizes” (edgeR) vs “size factors” (DESeq2)
Challenges for RNA-seq data analysis

- Choice of statistical distribution
- Normalization between samples
- Few samples -> difficult to estimate parameters (e.g., variance)
- High dimensionality (many genes) -> many tests
Example:
estimate variance of
normally distributed
variable

True value = 3
Modeling counts

- **Negative binomial distribution**
  
  - $\text{var}(X) = \mu + \theta \mu^2$
  
  - $\theta =$ dispersion
  
  - $\sqrt{\theta} =$ "biological coefficient of variation"
  
  - Allows mRNA proportions to vary across samples
  
  - Captures variability across biological replicates better
Shrinkage dispersion estimation

- Take advantage of the large number of genes
- Shrink the gene-wise estimates towards a center value defined by the observed distribution of dispersions across
  - all genes ("common" dispersion estimate)
  - genes with similar expression ("trended" dispersion estimate)
Shrinkage dispersion estimation

mean of normalized counts

dispersion
Shrinkage dispersion estimation

- mean of normalized counts

- dispersion

- gene-est
- fitted
- final
Shrinkage dispersion estimation

![Shrinkage dispersion estimation graph](image-url)
Challenges for RNA-seq data analysis

- Choice of statistical distribution
- Normalization between samples
- Few samples -> difficult to estimate parameters (e.g., variance)
- High dimensionality (many genes) -> many tests
Independent filtering

- Idea: filter out genes that have little chance of showing significance (without looking at the test results!!!)
- Improves detection power for remaining genes (fewer tests - less strict correction for multiple testing)
- For RNA-seq, typically filter based on expression
Independent filtering

- DESeq2:
  - filters based on the average normalized counts, using an optimized threshold.
  - p-values for excluded genes are set to NA in results

- edgeR:
  - manual filtering before applying test
  - all remaining genes are tested, and get a p-value
Testing against a threshold

• By default, the null hypothesis is that the log-fold change ($\beta$) between conditions is 0

• Both edgeR and DESeq2 can test more general null hypothesis, e.g. $|\beta| \leq 1$

• Useful if very small fold changes are not of interest

• Note that this is not the same as setting both a p-value and a fold change threshold on the regular test results!
Making use of the transcript abundances
Impact of differential isoform usage on gene-level counts

T1  
length = L

T2  
length = 2L

sample 1  
sample 2
Impact of differential isoform usage on gene-level counts

T1

T2

length = \( L \)

length = \( 2L \)

sample 1

sample 2
Impact of differential isoform usage on gene-level counts

<table>
<thead>
<tr>
<th>Gene</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>150</td>
<td>150</td>
</tr>
</tbody>
</table>

150 reads
Average transcript lengths

**T1**  
length = \( L \)

**T2**  
length = \( 2L \)

\[
ATL_{g1} = 1 \cdot L + 0 \cdot 2L = L
\]

\[
ATL_{g2} = 0 \cdot L + 1 \cdot 2L = 2L
\]
Average transcript lengths

T1  \[\text{length} = L\]

T2  \[\text{length} = 2L\]

\[ATL_{g1} = 0.75 \cdot L + 0.25 \cdot 2L = 1.25L\]

\[ATL_{g2} = 0.5 \cdot L + 0.5 \cdot 2L = 1.5L\]
Average transcript lengths

\[
ATL_{g1} = 0.75 \cdot L + 0.25 \cdot 2L = 1.25L
\]

\[
ATL_{g2} = 0.5 \cdot L + 0.5 \cdot 2L = 1.5L
\]

weights obtained from transcript TPM estimates
Offsets ("scaling factors")

\[ C_{ij} \sim NB(\mu_{ij} = s_{ij} q_{ij}, \theta_i) \]

- Extend scaling factor for given sample and gene to include the average length of the transcripts from the gene that are present in the sample.
Offsets ("average transcript lengths")

- Similar to correction factors for library size, but sample- and gene-specific
- Transcript abundance levels (TPMs) can be obtained from (e.g.) Salmon or kallisto
- Average transcript length for gene $g$ in sample $s$:

$$ATL_{gs} = \sum_{i\in g} \theta_{is} \bar{\ell}_{is}, \quad \sum_{i\in g} \theta_{is} = 1$$

$\bar{\ell}_{is} = \text{effective length of isoform } i \text{ (in sample } s)$
$\theta_{is} = \text{relative abundance of isoform } i \text{ in sample } s$