DESeq Model For RNA-seq Data

Biostatistics and Bioinformatics

June 22, 2021

Disclaimer

- DESeq has many limitations, not guaranteed to be the best
- There are alternative methods, each having their own limitations
- DESeq has nicely written R package
- DESeq is widely popular

[HTML] Moderated estimation of fold change and dispersion for RNA-seq data with **DESeq2**

MI Love, W Huber, S Anders - Genome biology, 2014 - genomebiology.biomedcentral.com

In comparative high-throughput sequencing assays, a fundamental task is the analysis of count data, such as read counts per gene in RNA-seq, for evidence of systematic changes across experimental conditions. Small replicate numbers, discreteness, large dynamic range …

☆ 99 Cited by 23851 Related articles All 30 versions »

Format of RNA-Seq Data

expid	CNAG_00001	CNAG_00002	CNAG_00003	CNAG_00004	CNAG_00005
<chr>	<int>	<int>	<int>	<int>	<int>
1_2019_P_M1_S1_L001_ReadsPerGene.out.tab	0	35	48	223	5
1_2019_P_M1_S1_L002_ReadsPerGene.out.tab	0	43	46	227	7
1_2019_P_M1_S1_L003_ReadsPerGene.out.tab	0	46	49	232	8
1_2019_P_M1_S1_L004_ReadsPerGene.out.tab	0	34	58	222	2

- RNA-Seq data are counts (not continuous real numbers)
- The total number of read counts varies across samples due to technical reasons

DESeq Data Format

K_{ij}	Sample 1	Sample 2	Sample 3
Gene 1	K_{11}	K_{12}	K_{13}
Gene 2	K_{21}	K_{22}	K_{23}
Gene 3	K_{31}	K_{32}	K_{33}
Gene 4	K_{41}	K_{42}	K_{43}
Gene 5	K_{51}	K_{52}	K_{53}

	Condition	Coded Condition	Gender	Coded Gender	Age
Sample 1	Untreated	0	Female	0	54
Sample 2	Treated	1	Male	1	67
Sample 3	Untreated	0	Male	1	39

Modeling RNA-Seq Data

Questions of interest:

- Do gene expression levels differ between two (or more) conditions?
- How are demographic/clinical/other variables associated with gene expression levels?

Connecting RNA-Seq read counts with features

- RNA-Seq counts (or expression level transformed from counts) is a perturbed version of true population mean
- The perturbation is described using a **distribution**
- The **mean parameters** of the distribution are connected to **a function of covariates**, such as linear combinations

Let's first focus on the study of **a given single gene**

Linear Regression Example

Connecting RNA-Seq read counts with features

- RNA-Seq counts (or expression level transformed from counts) is a perturbed version of true population mean
- The perturbation is described using a **distribution**
- The **mean parameters** of the distribution are connected to **a function of covariates**, such as linear combinations

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

or equivalently

$$Y_i \sim N(\mu_i, \sigma^2), \quad \mu_i = \beta_0 + \beta_1 X_i$$

Linear Regression

- Notation:
  - Y: **response/outcome/dependent variables**, continuous
  - $X_1, \ldots, X_p$: **predictors/covariates/independent variables**, can be continuous, count, categorical, etc.
- Goal: study the association between Y and $X_1, \ldots, X_p$.
- Model setting:

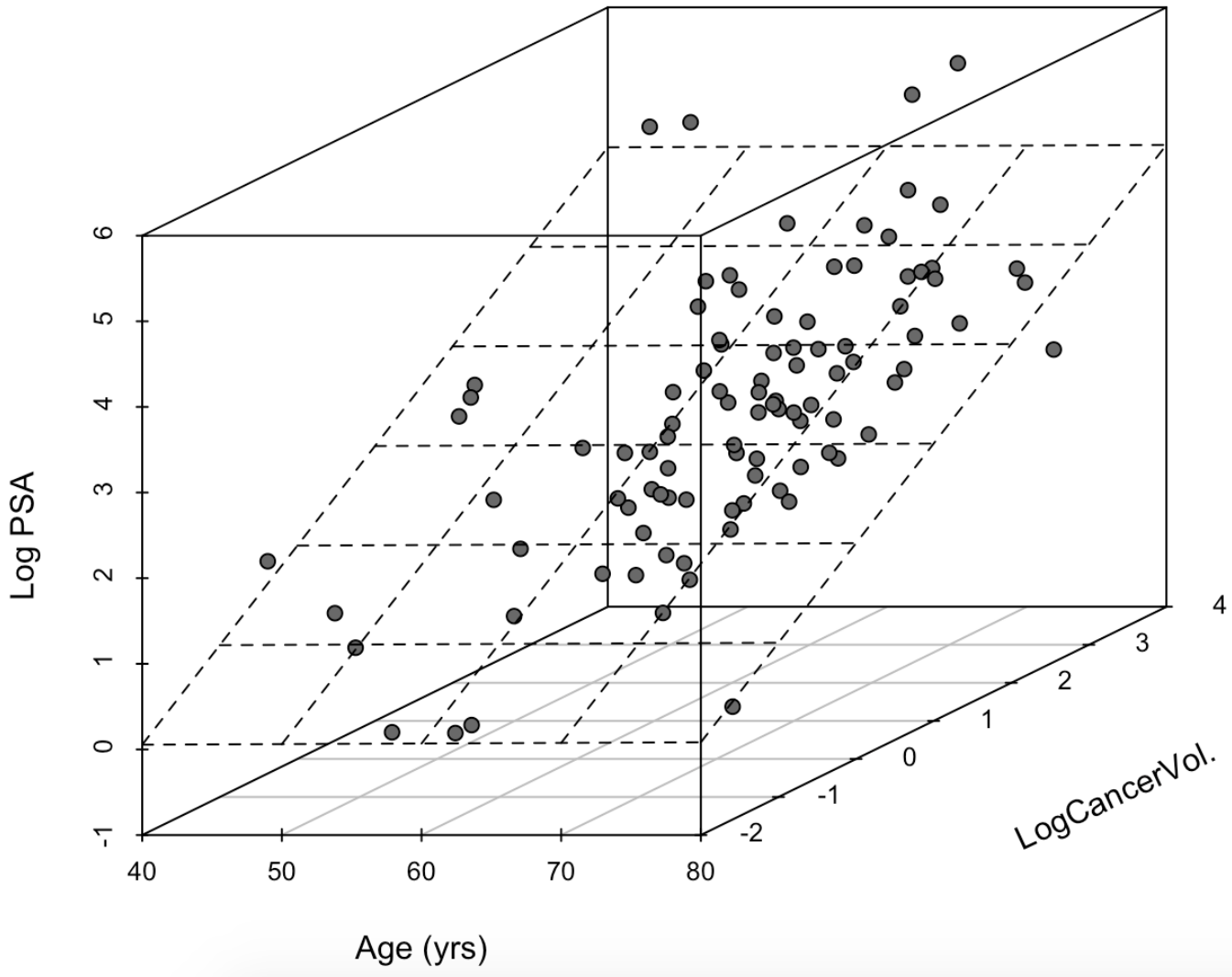
$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

- Mean and variance of Y conditional on $X_1, \ldots, X_p$

$$\mu = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p, \quad \sigma^2$$

- The magic here is incorporation of predictor information in μ
- Estimation of all parameters is achieved using MLE

Linear Regression Illustration



Linear Regression: One Predictor Example

- Suppose for now, the expression of a gene is measured in continuous scale.
- Y_i is the observed 'expression' of a gene from Sample i
- X_i denotes condition of Sample i: $X_i = 0$ for untreated, $X_i = 1$ for treated.
- Observed 'expression' $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$
- Population mean of Y_i conditional on X_i: $\mu_i = \beta_0 + \beta_1 X_i$
- $\mu_i = \beta_0$ for untreated, $\mu_i = \beta_0 + \beta_1$ for treated.
- Meaning of β_1: the **expected increase** of 'expression' when condition is switched from untreated to treated
- Meaning of β_0: the expected 'expression' when untreated
- σ^2: **nuissance parameter**, not of interest but needed in the estimation of CI

Linear Regression: Two Predictors Example

- Y_i is the observed 'expression' of a gene from Sample i
- X_{i1} denotes the condition of Sample i: $X_{i1} = 0$ for untreated, $X_{i1} = 1$ for treated.
- X_{i2} denotes age of Sample i
- Observed 'expression': $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \epsilon_i$
- Population mean of Y_i conditional on X_{i1} and X_{i2}: $\mu_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}$

Linear Regression: Two Predictors Example

- Mean of Y_i conditional on X_{i1} and X_{i2}:
  $\mu_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}$
- $\mu_i = \beta_0 + 40\beta_2$ for an untreated 40-year-old,
  $\mu_i = \beta_0 + \beta_1 + 40\beta_2$ for a treated 40-year-old,
  $\mu_i = \beta_0 + \beta_1 + 41\beta_2$ for a treated 41-year-old,
- Meaning of β_1: with age unchanged, the expected increase of 'expression' if the individual is switched from untreated to treated
- Meaning of β_2: with condition unchanged, the expected increase of 'expression' if age is increased by one unit.
- Meaning of β_0: expected 'expression' of an untreated 0-year-old, a nuisance parameter

Modeling RNA-Seq Counts

- RNA-Seq counts is a perturbed version of true population mean
- The perturbation is described using a **distribution**
- The **mean parameters** of the distribution are connected to **a function of covariates**, such as linear combinations
- Linear regresion uses Gaussian distribution, which is continuous
- We need distribution for counts

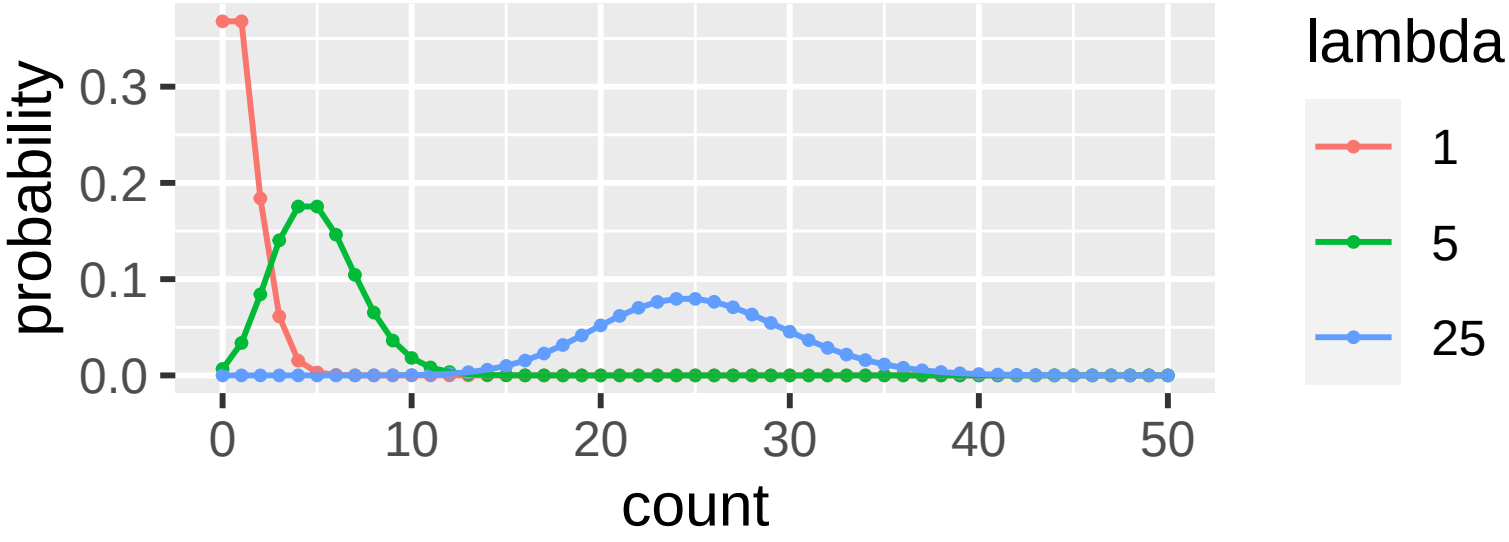
Distributions for Count Data

Modeling the read count of a **given gene**:

- Poisson distribution
- Binomial distribution
- Negative Binomial distribution
- Others not discussed here: geometric distribution, beta-binomial distribution, etc.
- Continuous distributions can be used after transformation of count data

Poisson Distribution

- Used to model the count of occurrence of events
- Classical application: the number of patient arriving at an emergency room within a day
- Mean $\mu = \lambda$ and Variance $\sigma^2 = \lambda$
- Do highly expressed genes also have high variation in expression?

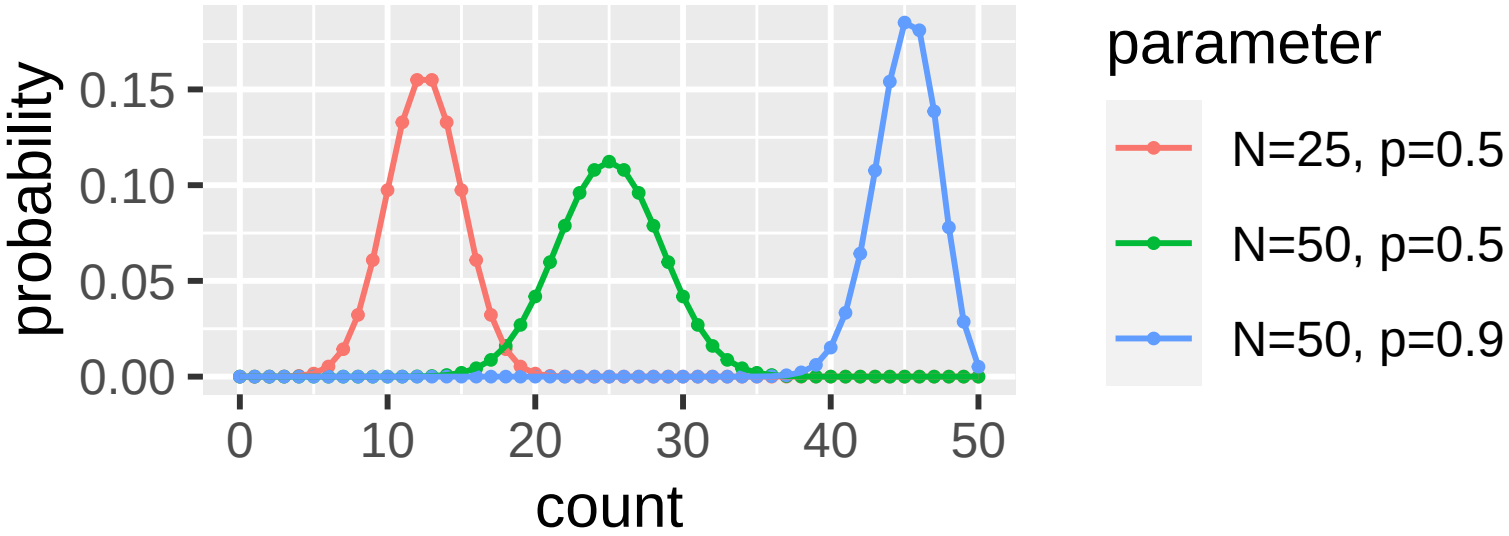


Binomial Distribution

- Used to model the number of $(+)$s in a sequence of N identical independent experiments with $(+)/(-)$ outcomes
- Classical application: the number of heads in N flips of a coin when probability of head is p
- Mean and variance:

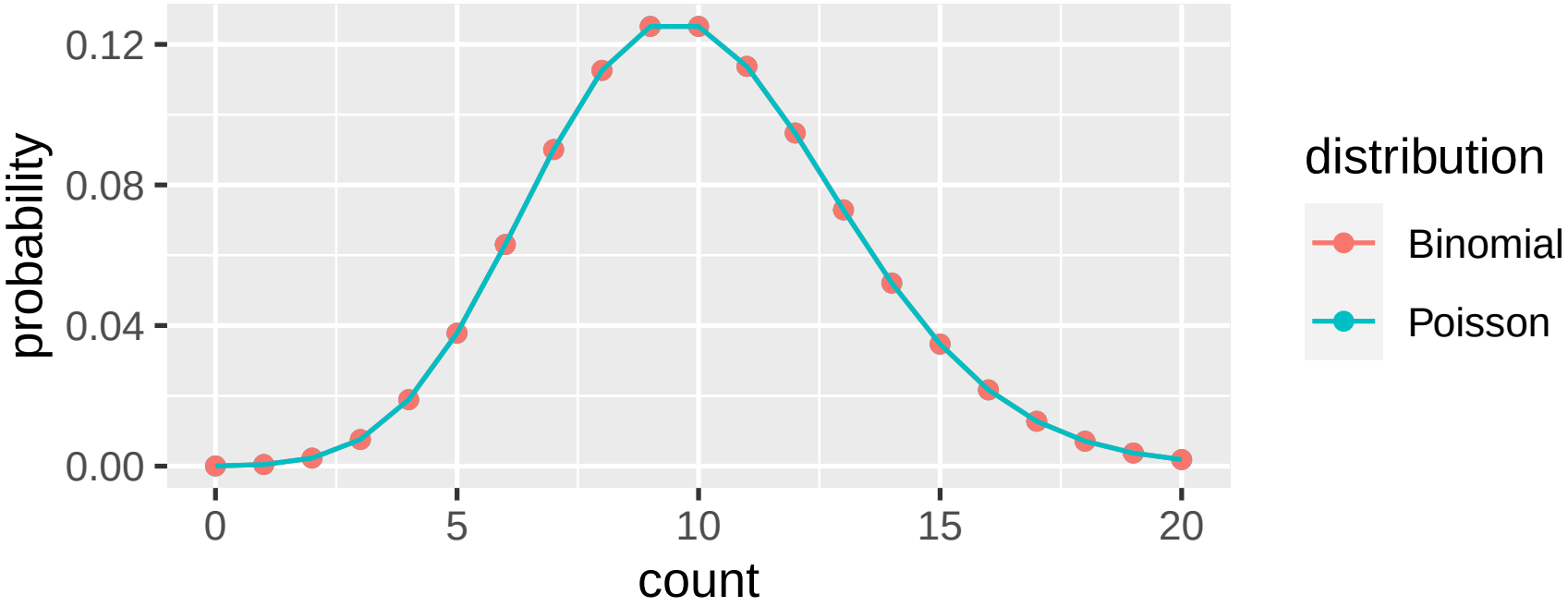
$$\mu = Np, \quad \sigma^2 = Np(1-p)$$

- What N to choose to model RNA-Seq read counts?
- Does variance σ^2 have any freedom once mean μ is fixed?



Connection between Poisson and Binomial Distribution

- As $N \to \infty$, binomial distribution converges to Poisson distribution
- Consider tossing a coin for $N = 10^6$ times with a small success rate $p = 10^{-5}$, i.e. Binomial distribution with $N = 10^6$ and $p = 10^{-5}$.
- Mean: $\mu = Np = 10$ and Variance $\sigma^2 = Np(1-p) = 9.9999$
- It is very close to Poisson distribution with mean $\lambda = 10$



Negative Binomial Distribution

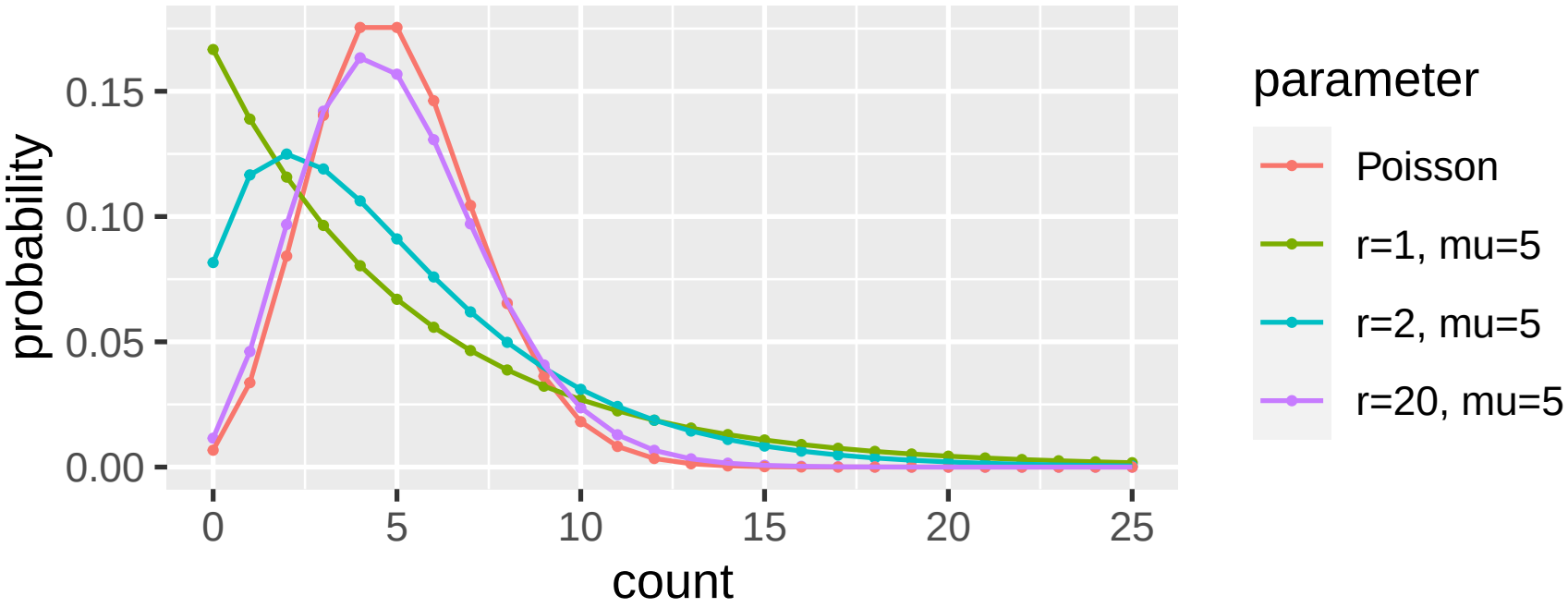
- Used to model how many $(-)$s before getting $r > 0$ $(+)$s.
- Example: the number of male birth before the rth female birth when probability of female birth is p
- It is also connected to Poisson (theories not discussed here), as a "Poisson with extra variance"
- Mean and variance:

$$\mu = r\frac{1-p}{p}, \quad \sigma^2 = r\frac{1-p}{p^2} = \mu(1 + \mu \underbrace{/r}_{\alpha})$$

- $\alpha = 1/r$ is often called the **overdispersion parameter**
- As $r \to \infty$ or $\alpha \to 0$, negative binomial becomes Poisson.

Negative Binomial Distribution (continued)

- The variance is $\mu(1+\alpha\mu) = \mu(1+\mu/r)$
- The flexibility of having varying variance without changing mean
- Larger $\alpha \Leftrightarrow$ smaller $r \Leftrightarrow$ more overdispersion $\Leftrightarrow$ heavier right tail of PMF $\Leftrightarrow$ less like Poisson

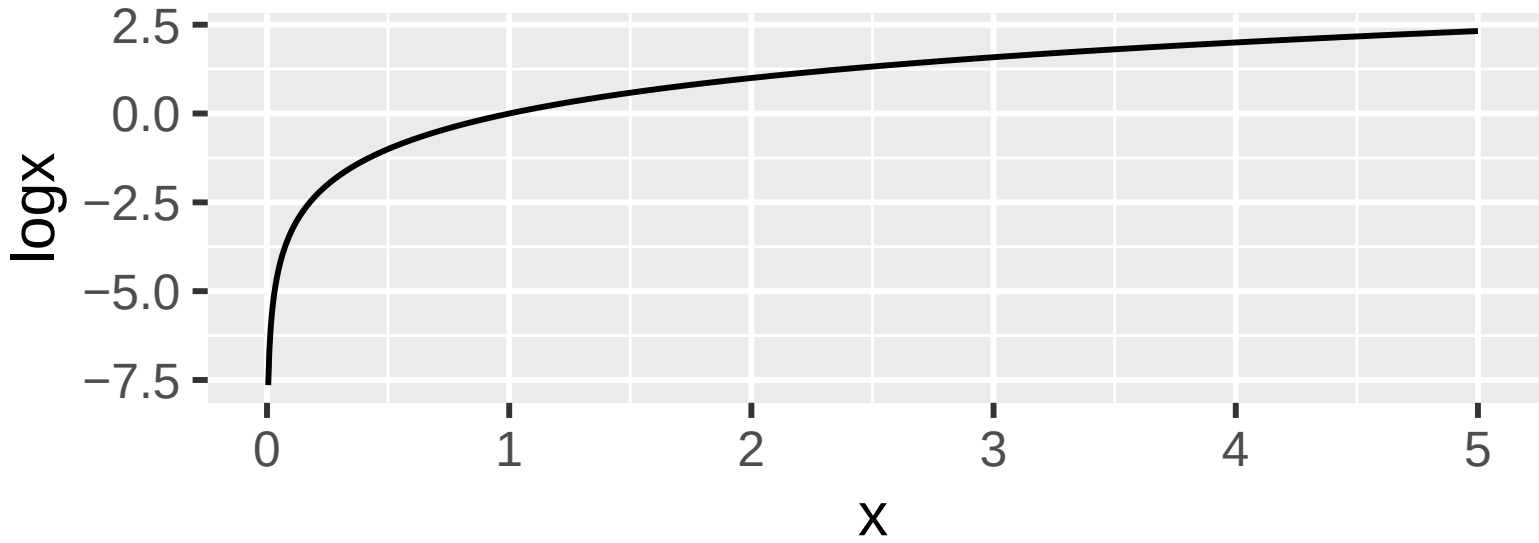


Negative Binomial Regression

- Goal: study the association between count variable K and $X_1, \ldots, X_p$.
- Model setting: K follows NB with mean and variance below

$$\mu = 2^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}, \quad \sigma^2 = \mu(1+\alpha\mu)$$

- $\log_2(\mu) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$, why $\log_2$?
- $\log_2(\cdot)$ is called the **link function**, which links the predictors with μ (not K since K is random)



NB Regression: One Predictor Example

- K_i is the number of reads mapped to a given gene from Sample i
- X_i denotes the condition of Sample i: $X_i = 0$ for untreated, $X_i = 1$ for treated
- Population mean of Y_i conditional on X_i: $\mu_i = 2^{\beta_0 + \beta_1 x_i}$
- $\mu_i = 2^{\beta_0}$ for parental, $\mu_i = 2^{\beta_0 + \beta_1}$ for UV2.
- Meaning of β_1: gene expression level is expected to be **multiplied by** 2^{β_1} when condition is switched from untreated to treated
  - β_1 is the **log 2 fold change (LFC)**
  - What does it mean when $\beta_1 < 0, = 0, > 0$?
- β_0 and α are nuisance parameters
- Dispersion parameter α will affect the length of CIs

NB Regression: Two Predictors Example

- K_i is the number of reads mapped to a given gene from Sample i
- X_{i1} denotes the condition of Sample i: $X_{i1} = 0$ for untreated, $X_{i1} = 1$ for treated
- X_{i2} denotes age of Sample i
- Mean of Y_i conditional on X_{i1} and X_{i2}: $\mu_i = 2^{\beta_0+\beta_1 X_{i1}+\beta_2 X_{i2}}$
- $\mu_i = 2^{\beta_0+40\beta_2}$ for an untreated 40-year-old, $\mu_i = 2^{\beta_0+\beta_1+40\beta_2}$ for a treated 40-year-old, $\mu_i = 2^{\beta_0+\beta_1+41\beta_2}$ for a treated 41-year-old,
- Meaning of β_1: with age unchanged, the expression is expected to be multiplied by 2^{β_1} if the individual is switched from untreated to treated
- Meaning of β_2: with condition unchanged, the expression is expected to be multiplied by 2^{β_2} if age is increased by one unit.
- β_0 and α are nuisance parameters
- Dispersion parameter α will affect the length of CIs

Changing the Coding of Variables

- We have used $X_i = 1$ for treated and $X_i = 0$ for untreated.
  - $\mu_i = 2^{\beta_0 + \beta_1 X_i}$
  - $\mu_i = 2^{\beta_0}$ for untreated, $\mu_i = 2^{\beta_0 + \beta_1}$ for treated.
  - LFC from untreated to treated β_1
- If we change the coding into $\widetilde{X}_i = 1$ for untreated and $\widetilde{X}_i = 0$ for treated, i.e. $\widetilde{X}_i = 1 - X_i$
  - $$\log_2(\mu_i) = \beta_0 + \beta_1(1 - \widetilde{X}_i) = \underbrace{(\beta_0 + \beta_1)}_{\beta_0^*} \underbrace{-\beta_1}_{\beta_1^*} \widetilde{X}_i$$
  - $\mu_i = 2^{\beta_0^*}$ for treated, $\mu_i = 2^{\beta_0^* + \beta_1^*}$ for untreated.
  - LFC from untreated to treated: $-\beta_1^* = \beta_1$
- When the coding of a binary variable is changed, the sign of coefficient changed!

Now let's move on to study **multiple genes**

DESeq Data Format

K_{ij}	Sample 1	Sample 2	Sample 3
Gene 1	K_{11}	K_{12}	K_{13}
Gene 2	K_{21}	K_{22}	K_{23}
Gene 3	K_{31}	K_{32}	K_{33}
Gene 4	K_{41}	K_{42}	K_{43}
Gene 5	K_{51}	K_{52}	K_{53}

	Condition	Coded Condition	Gender	Coded Gender	Age
Sample 1	Untreated	0	Female	0	54
Sample 2	Treated	1	Male	1	67
Sample 3	Untreated	0	Male	1	39

DESeq Notation and Model Setting

- K_{ij} denotes the observed number of reads mapped to Gene i for sample j, $i = 1, \ldots, m, j = 1, \ldots, n$
- K_{ij} follows NB with
  - Mean μ_{ij} (indexed by Gene i and Sample j)
  - Dispersion parameter α_i (indexed by the Gene i)
- The mean is assumed to be $\mu_{ij} = s_j q_{ij}$ where
  - $\log_2(q_{ij}) = \beta_{i0} + \beta_{i1} X_{j1} + \cdots + \beta_{ip} X_{jp}$
  - s_j is a Gene j specific normalization constant accounting for varying sequencing depth

DESeq: One Predictor Example

- X_j denotes condition of Sample j, $X_j = 0$ for untreated, $X_j = 1$ for treated
- K_{ij} denotes the observed number of reads mapped to Gene i for sample j
- K_{ij} follows NB with
  - Mean $\mu_{ij} = s_j 2^{\beta_{i0}+\beta_{i1}x_j}$
  - Dispersion parameter α_i
- Meaning of β_{i1}: expression of Gene i is expected to be **multiplied by** $2^{\beta_{i1}}$ when condition is switched from untreated to treated
- β_{i1} is the LFC of Gene i
- Why does β_{i1} have index i?
- Why does X has index j but not i?
- Why does s_j has index j but not i?
  - Implicated assumption: within Sample j, the normalization parameter is constant across all the genes
- Why does α_i has index i but not j?

DESeq: Two Predictors Example

- X_{j1} denotes condition of Sample j, $X_{j1} = 0$ for untreated, $X_{j1} = 1$ for treated
- X_{j2} denotes age of Sample j
- K_{ij} denotes the observed number of reads mapped to Gene i for Sample j
- K_{ij} follows NB with
  - Mean $\mu_{ij} = s_j 2^{\beta_{i0}+\beta_{i1}x_{j1}+\beta_{i2}x_{j2}}$
  - Dispersion parameter α_i
- Meaning of β_{i1}: when age stays unchanged, expression of Gene i is expected to be **multiplied by** $2^{\beta_{i1}}$ when condition is switched from untreated to treated
- Meaning of β_{i2}: when condition stays unchanged, expression of Gene i is expected to be **multiplied by** $2^{\beta_{i2}}$ when age is increased by one unit

DESeq Parameter Summary

- The main parameters of interest
  - m parameters on the effect of the 1st predictor

$$\beta_{11}, \ldots, \beta_{m1}$$

  - m parameters on the effect of the 2nd predictor

$$\beta_{12}, \ldots, \beta_{m2}$$

  - more parameters for more predictors
- The unknown nuisance parameters are
  - The m gene specific intercepts

$$\beta_{10}, \ldots, \beta_{m0}$$

  - The n sample specific normalization constants

$$s_1, \ldots, s_n$$

  - The m gene specific dispersion parameters

$$\alpha_1, \ldots, \alpha_m$$

DESeq Parameter Estimation

- If s_j and α_i are known, $\beta_{i0}, \beta_{i1}, \ldots$ can be estimated using MLE
- The DESeq authors propose to estimate the normalization constant for sample j as

$$s_j = \text{median} \frac{K_{ij}}{K_i^R}$$

where

$$K_i^R = \left(\prod_{j=1}^{m} K_{ij} \right)^{\frac{1}{m}}$$

- Here K_i^R is the geometric mean of $K_{i1}, \ldots, K_{in}$ (the n counts for gene i)
- The median is taken over all m genes for which K_i^R is positive

DESeq Parameter Estimation

- A key issue in using the NB model is proper handling of the gene specific dispersion parameters

$$\alpha_1, \ldots, \alpha_m$$

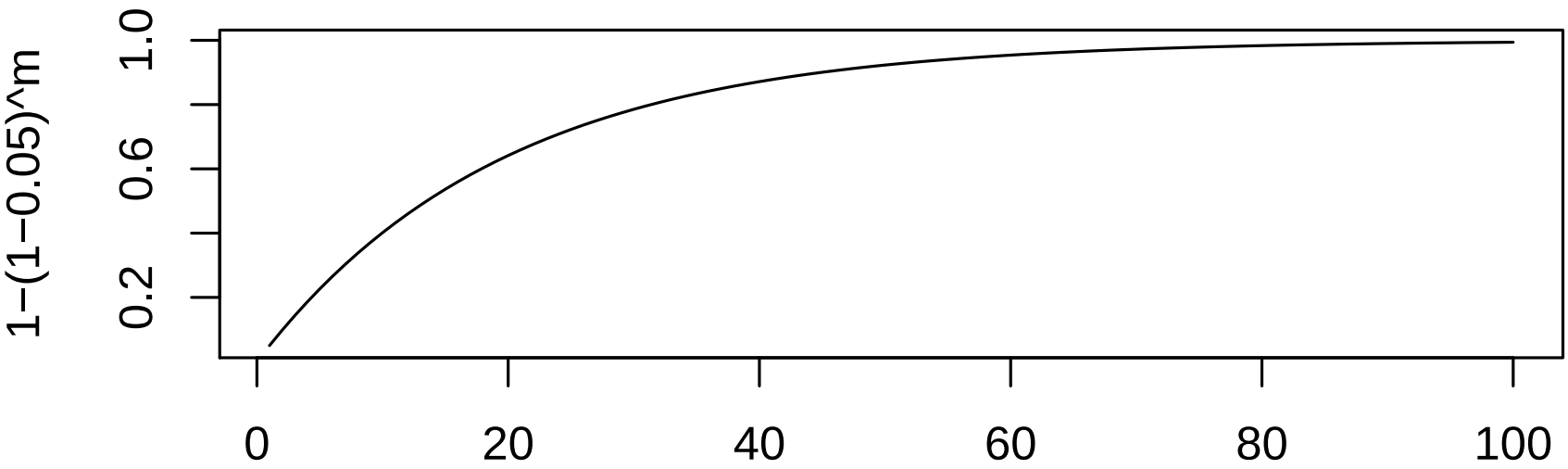
- The estimation of the dispersion parameter is a challenging task
- DESeq2 assumes that α_i is random following a normal distribution
- The results are sensitive to the estimates
- One of the key differences between DESeq2 and DESeq is the approach taken to estimate these nuisance parameters

DESeq Hypotheses Testing

- Use one-predictor example: $\mu_{ij} = s_j 2^{\beta_{i0}+\beta_{i1}x_{j1}}$ where X_{j1} denotes condition untreated/treated
- Gene-level hypotheses:
  - Does treatment affect the expression of Gene i?
  - Null: $H_{0i} : \beta_{i1} = 0$
  - Alternative: $H_{ai} : \beta_{i1} \neq 0$
- Global hypotheses:
  - Does treatment affect the expression of any genes?
  - Null: $H_0 : \beta_{11} = \beta_{21} = \cdots = \beta_{m1} = 0$, or equivalently $H_{01}, \ldots, H_{0m}$ are all true
  - Alternative: H_a : at least one of $\beta_{i1} \neq 0$, or equivalently at least one of $H_{a1}, \ldots, H_{am}$ is true

Multiple Testing

- Suppose we test for differential expression of m independent genes between the two cell lines
- Denote $H_{0i} : \mu_{i1} = \mu_{i2}$ vs $H_{1i} : \mu_{i1} \neq \mu_{i2}$ as the hypotheses for Gene i,
- Suppose none of the m genes are differentially expressed, H_{0i} is true for all genes
- The probability of not rejecting each H_{0i} is $1 - \alpha$
- The probability of making no type I error is $(1 - \alpha)^m$
- The probability of making any type I error is $1 - (1 - \alpha)^m$



Counting the Correct and Wrong Decisions

- m: total number of genes
- m_0: number of genes without differential expression
- m_1: number of differentially expressed genes
- R: number of genes rejecting H_{0i} according to the decision rule
- A: number of genes failing to reject H_{0i}

	H_{0i} is true	H_{1i} is true	Total
H_{0i} is not rejected	A_0 (TN)	A_1 (FN)	A
H_{0i} is rejected	R_0 (FP)	R_1 (TP)	R
	m_0	m_1	m

Two Error Rate for Multiple Testing

	H_{0i} is true	H_{1i} is true	Total
H_{0i} is not rejected	A_0 (TN)	A_1 (FN)	A
H_{0i} is rejected	R_0 (FP)	R_1 (TP)	R
	m_0	m_1	m

- Family-wise error rate (FWER): the probability of making at least one type I error, i.e. $Pr(R_0 \geq 1)$
- False discovery rate (FDR), i.e. the expected proportion of wrong rejections (type I errors) over all rejection of H_{0i}, i.e. expectation of R_0/R, or more formally $E(\frac{R_0}{R}|R > 0) * Pr(R > 0)$
- When $m_0 = m$ (no gene is differentially expressed), FWER=FDR

Simulated Example of FWER and FDR

- Suppose the first 10 genes among 100 genes are differentially expressed, and the rest are not.
- Let's try to test each individual gene without correction for multiple testing

```
set.seed(2021)
n <- 20 # sample size
rejected <- rep(FALSE, 100)
for (g in 1:10){
  y1 <- rnorm(n); y2 <- rnorm(n)+1
  rejected[g] <- t.test(y1,y2)$p.value<0.05
}
for (g in 11:100){
  y1 <- rnorm(n); y2 <- rnorm(n)
  rejected[g] <- t.test(y1,y2)$p.value<0.05
}
ind_rejected <- which(rejected)
print(ind_rejected)
```

```
##  [1]  2  3  6  7  8  9 10 11 19 63 93
```

```
# family-wise error: making any type I error
FWE <- any(ind_rejected>10)
# false discovery proportion: proportion of wrong rejection in all rejections
FDP <- sum(ind_rejected>10)/max(1,length(ind_rejected))
print(c(FWE,FDP))
```

```
## [1] 1.0000000 0.3636364
```

Simulated Example of FWER and FDR

- Suppose the first 10 genes among 100 genes are differentially expressed, and the rest are not.
- Let's try to test each individual gene without correction for multiple testing

```r
set.seed(2021)
n <- 20 # sample size
nsim <- 1000 # number of simulation rounds
ngene <- 100; ndiff <- 10
FWE <- rep(FALSE, nsim); FDP <- rep(0, nsim)
for (i in 1:nsim){
  rejected <- rep(FALSE, 100)
  for (g in 1:ndiff){
    y1 <- rnorm(n); y2 <- rnorm(n)+1
    rejected[g] <- t.test(y1,y2)$p.value<0.05
  }
  for (g in (ndiff+1):ngene){
    y1 <- rnorm(n); y2 <- rnorm(n)
    rejected[g] <- t.test(y1,y2)$p.value<0.05
  }
  ind_rejected <- which(rejected)
  # family-wise error: making any type I error
  FWE[i] <- any(ind_rejected>ndiff)
  # false discovery proportion: proportion of wrong rejection in all rejections
  FDP[i] <- sum(ind_rejected>ndiff)/max(1,length(ind_rejected))
}
# family wise error rate (FWER), false discovery rate (FDR)
print(c(mean(FWE), mean(FDP)))
```

```
## [1] 0.9860000 0.3221419
```

Correction for Multiple Testing

- To achieve FWER$< \alpha$ or FDR$< \alpha$, rejection of each H_{0i} should be harder than the original decision rule that controls for type I error of each H_{0i}
- Method to achieve FWER$< \alpha$
  - Bonferroni's method: reject H_{0i} if its pvalue$< \frac{\alpha}{m}$. Easy to calculate, doesn't require independence among genes, can be super conservative
- Method to achieve FDR$< \alpha$
  - Benjamini–Hochberg method: assumes independence among genes, generally less conservative than Bonferroni's method
  - Benjamini-Yekutieli method: allows for positive dependence, but more conservative in making rejections

Adjusted p-values

- After the correction of multiple testing, the adjusted p-value for each individual H_{0i} is larger. (How about the length of CI?)
- The adjusted p-value of each gene changes if you change the set of genes to test
  - In general, the more genes you test, the more correction you need
- q-value: the adjusted p-value to control pFDR (a slightly different definition of FDR)

Simulated Example of FWER and FDR (Bonferroni Correction)

- Suppose the first 10 genes among 100 genes are differentially expressed, and the rest are not.
- Let's try the Bonferroni's method

```
set.seed(2021)
n <- 20 # sample size
nsim <- 1000 # number of simulation rounds
ngene <- 100; ndiff <- 10 # experiment the codes by changing these values
FWE <- rep(FALSE, nsim); FDP <- rep(0, nsim)
for (i in 1:nsim){
  rejected <- rep(FALSE, 100)
  for (g in 1:ndiff){
    y1 <- rnorm(n); y2 <- rnorm(n)+1
    rejected[g] <- t.test(y1,y2)$p.value<0.05/ngene
  }
  for (g in (ndiff+1):ngene){
    y1 <- rnorm(n); y2 <- rnorm(n)
    rejected[g] <- t.test(y1,y2)$p.value<0.05/ngene
  }
  ind_rejected <- which(rejected)
  # family-wise error: making any type I error
  FWE[i] <- any(ind_rejected>ndiff)
  # false discovery proportion: proportion of wrong rejection in all rejections
  FDP[i] <- sum(ind_rejected>ndiff)/max(1,length(ind_rejected))
}
# family wise error rate (FWER), false discovery rate (FDR)
print(c(mean(FWE), mean(FDP)))
```

```
## [1] 0.03600000 0.01076349
```