Differential Expression Analysis of RNA Sequencing Data

Ryan C. Thompson

August 15, 2016

Outline

- What does RNA-seq measure, and how?
- The raw data: DNA sequence reads
- Getting from reads to counts
- Basic analysis strategy: linear models
- Normalization of RNA-seq counts
- Heteroskedasticity!
- Sharing information between genes
- Multiple testing and FDR

Introduction

What is RNA sequencing?

What kind of data does it produce?

What is RNA sequencing?

What kind of data does it produce?

- RNA sequence reads!
- Reads are sampled randomly from the population of RNA transcripts in a sample.

What is RNA sequencing?

What kind of data does it produce?

- RNA sequence reads!
- Reads are sampled randomly from the population of RNA transcripts in a sample.

How can we measure gene expression levels using RNA sequences?

What is RNA sequencing?

What kind of data does it produce?

- RNA sequence reads!
- Reads are sampled randomly from the population of RNA transcripts in a sample.

How can we measure gene expression levels using RNA sequences?

- Align reads to the transcriptome
- Count the reads that align uniquely to each gene
- A gene's count should be proportional its expression level

What is RNA sequencing?

What kind of data does it produce?

- RNA sequence reads!
- Reads are sampled randomly from the population of RNA transcripts in a sample.

How can we measure gene expression levels using RNA sequences?

- Align reads to the transcriptome
- Count the reads that align uniquely to each gene
- A gene's count should be proportional its expression level

Can we measure anything else?

What is RNA sequencing?

What kind of data does it produce?

- RNA sequence reads!
- Reads are sampled randomly from the population of RNA transcripts in a sample.

How can we measure gene expression levels using RNA sequences?

- Align reads to the transcriptome
- Count the reads that align uniquely to each gene
- A gene's count should be proportional its expression level

Can we measure anything else?

- Alternative splicing
- Discover novel splices/isoforms
- Genotype coding SNPs

What is RNA sequencing?

What kind of data does it produce?

- RNA sequence reads!
- Reads are sampled randomly from the population of RNA transcripts in a sample.

How can we measure gene expression levels using RNA sequences?

- Align reads to the transcriptome
- Count the reads that align uniquely to each gene
- A gene's count should be proportional its expression level

Can we measure anything else?

- Alternative splicing
- Discover novel splices/isoforms
- Genotype coding SNPs

(I won't be covering these, but just be aware that these are options)

What do “reads” look like?

- “FASTQ” format: contains both DNA sequence and quality values for each base

What do "reads" look like?

- "FASTQ" format: contains both DNA sequence and quality values for each base

```
@071112_SLXA-EAS1_s_7:5:1:817:345
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+071112_SLXA-EAS1_s_7:5:1:817:345
IIIIIIIIIIIIIIIIIIIIIIIIIIIII9IG9IC
@071112_SLXA-EAS1_s_7:5:1:801:338
GTTCAGGGATACGACGTTTGTATTTTAAGAATCTGA
+071112_SLXA-EAS1_s_7:5:1:801:338
IIIIIIIIIIIIIIIIIIIIIIIIIIIIII6IBI
```

What do "reads" look like?

- "FASTQ" format: contains both DNA sequence and quality values for each base

```
@071112_SLXA-EAS1_s_7:5:1:817:345
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+071112_SLXA-EAS1_s_7:5:1:817:345
IIIIIIIIIIIIIIIIIIIIIIIIIIIII9IG9IC
@071112_SLXA-EAS1_s_7:5:1:801:338
GTTCAGGGATACGACGTTTGTATTTTAAGAATCTGA
+071112_SLXA-EAS1_s_7:5:1:801:338
IIIIIIIIIIIIIIIIIIIIIIIIIIIIII6IBI
```

- Often compressed with gzip, e.g. `Sample_A.fastq.gz`

Here's the general idea

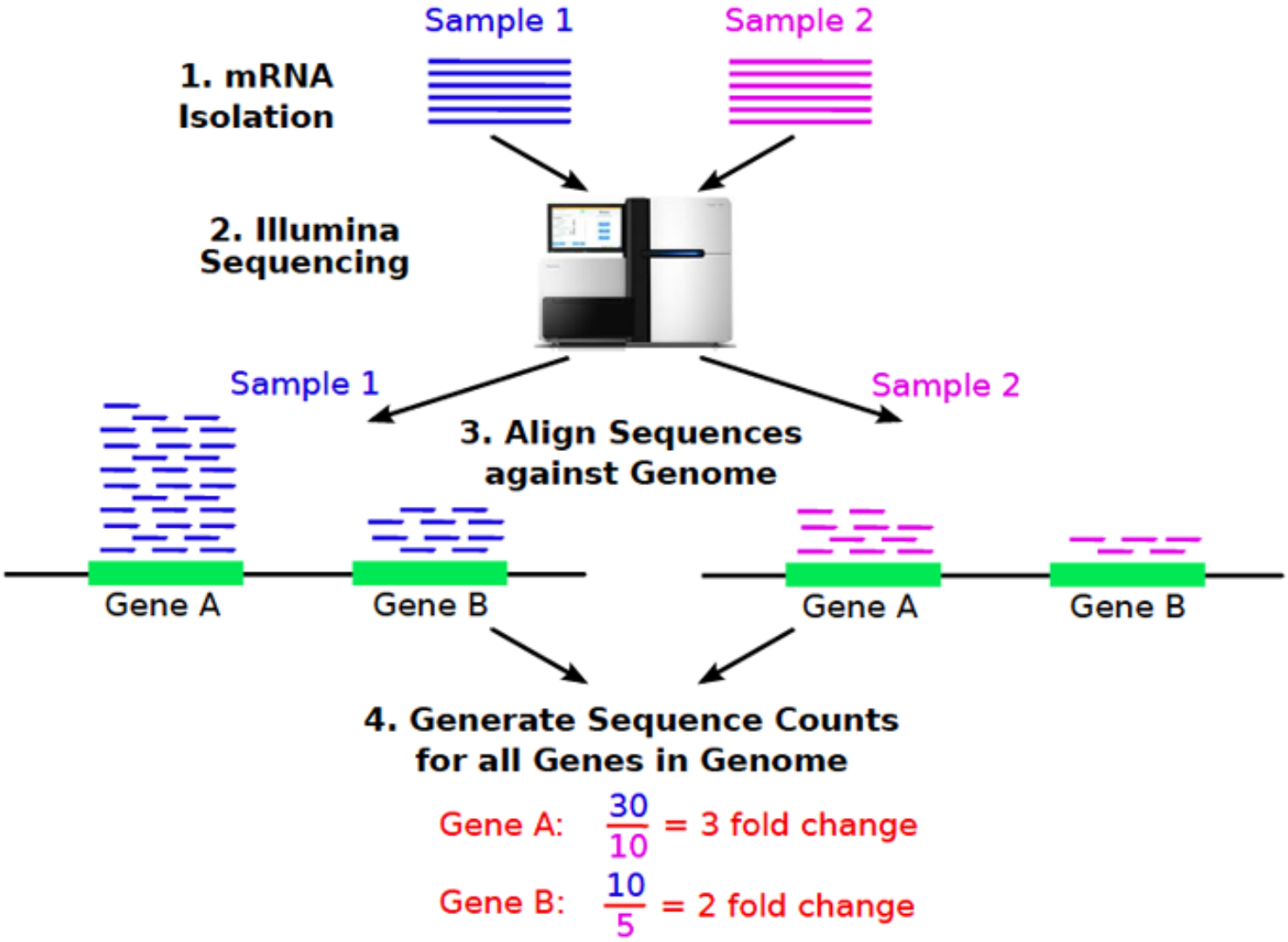

Figure 1: Basic RNA-seq data generation

Generating a count matrix from an RNA-seq experiment

- RNA-seq produces millions of read sequences
- We treat each read as a single observation of a gene, and assume that the abundance of the gene is proportional to how many times it is "observed".
- We **align** those reads to the matching sequence in the genome (or transcriptome), then we **count** the number of reads in each sample that align unambiguously to each gene.
- We discard ambiguous reads, so each count is an exact integer.
- There are more subtleties to counting reads (multi-mapping, alternative splicing, etc.). I'm not covering them here.
- The end result is a **table of genes X samples** with integer counts.

Aligning reads: accounting for splicing

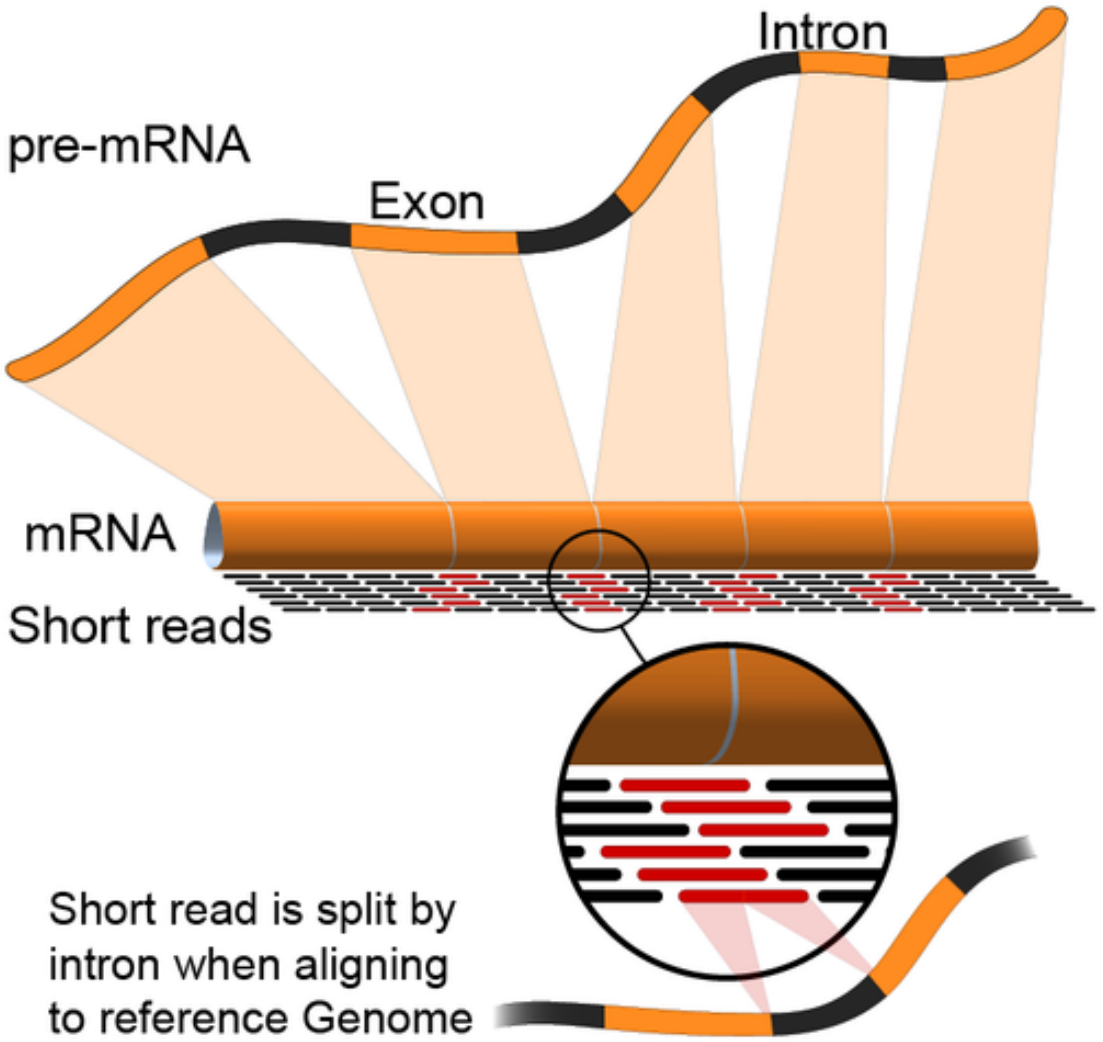

Figure 2: Split mapping of spliced reads

Statistical Analysis of RNA-seq Counts

Our basic strategy: linear models!

- First, we take the logarithm of the the counts, since the distribution after log transform is closer to a normal distribution (i.e. better for linear modeling)
- Then, for each gene we run:

```
summary(lm(log_count ~ covariates))
```

- Get a p-value for each gene

Our basic strategy: linear models!

- First, we take the logarithm of the the counts, since the distribution after log transform is closer to a normal distribution (i.e. better for linear modeling)
- Then, for each gene we run:

  ```
  summary(lm(log_count ~ covariates))
  ```

- Get a p-value for each gene

Well, that was easy. I'll see you this afternoon for the lab.

Our basic strategy: linear models!

- First, we take the logarithm of the the counts, since the distribution after log transform is closer to a normal distribution (i.e. better for linear modeling)
- Then, for each gene we run:

```
summary(lm(log_count ~ covariates))
```

- Get a p-value for each gene

Well, that was easy. I'll see you this afternoon for the lab.
Just kidding! This simple analysis has a bunch of issues that we need to fix if we want a valid analysis.

What's wrong with a simple linear model like this?

Questions to think about:

- Does the same count always equal the same expression? Is a count of 10 in sample A equal to a count of 10 in sample B?
- Are all counts equally precise? Which count is more precise, a 10 or 1000?
- Is our sample size large enough to do estimate parameters precisely?
- Can we use $P < 0.05$ as our significance threshold? How can we determine a better threshold?

What's wrong with a simple linear model like this?

Questions to think about:

- Does the same count always equal the same expression? Is a count of 10 in sample A equal to a count of 10 in sample B?
- Are all counts equally precise? Which count is more precise, a 10 or 1000?
- Is our sample size large enough to do estimate parameters precisely?
- Can we use $P < 0.05$ as our significance threshold? How can we determine a better threshold?

We'll see how we can embellish the standard linear model described earlier to address all of these problems.

Scaling Normalization for Count Data

Why normalize?

- Why do we have to normalize RNA-seq counts?
- What factors do we need to normalize for?
- What factors do we *not* need to normalize for?

CPM: Normalizing for sequencing depth

- The number of reads output by the sequencer for each sample is random
- Some samples receive more reads than others
- The total read count can vary by over 2-fold in many cases
- 10 counts out of 1 million is not the same as 10 counts out of 2 million
- Introduce “counts per million”, a.k.a. CPM

CPM: Normalizing for sequencing depth

- The number of reads output by the sequencer for each sample is random
- Some samples receive more reads than others
- The total read count can vary by over 2-fold in many cases
- 10 counts out of 1 million is not the same as 10 counts out of 2 million
- Introduce "counts per million", a.k.a. CPM

So, 10 counts is not the same in every sample, but 10 CPM is, right?

Example count table

Gene	Control	Treatment
Gene1	10	10
Gene2	10	10
Gene3	20	10
Gene4	20	10
Gene5	20	10
Gene6	20	10
Gene7	20	10
Gene8	20	10
Gene9	20	10
Gene10	40	10
Gene11	100	200
Total	**300**	**300**

Example CPM table

Gene	Control	Treatment	log2(T/C)
Gene1	33333	33333	0
Gene2	33333	33333	0
Gene3	66667	33333	-1
Gene4	66667	33333	-1
Gene5	66667	33333	-1
Gene6	66667	33333	-1
Gene7	66667	33333	-1
Gene8	66667	33333	-1
Gene9	66667	33333	-1
Gene10	133333	33333	-2
Gene11	333333	666667	+1

Example CPM table

Gene	Control	Treatment	log2(T/C)
Gene1	33333	33333	0
Gene2	33333	33333	0
Gene3	66667	33333	-1
Gene4	66667	33333	-1
Gene5	66667	33333	-1
Gene6	66667	33333	-1
Gene7	66667	33333	-1
Gene8	66667	33333	-1
Gene9	66667	33333	-1
Gene10	133333	33333	-2
Gene11	333333	666667	+1

Which gene(s) were affected by the treatment, and how were they affected?

TMM: Normalizing for compositional bias

- Instead of normalizing for total counts, normalize so that the average log fold change is zero
- We have lots of genes, so we make the average robust against outliers by throwing away the highest- and lowest-abundance genes.
- Also throw away the highest and lowest fold changes for the same reason
- Result: "Trimmed Mean of M-values" method, a.k.a. TMM (M-values are the term for log fold changes)
- Apply the normaliztion by modifying the total counts and then computing CPM using the modified totals

So how would we normalize this table with TMM?

Gene	Control	Treatment
Gene1	10	10
Gene2	10	10
Gene3	20	10
Gene4	20	10
Gene5	20	10
Gene6	20	10
Gene7	20	10
Gene8	20	10
Gene9	20	10
Gene10	40	10
Gene11	100	200
Total	300	300
TMM correction	$\sqrt{2}$	$1/\sqrt{2}$
Normalized total	424.26	212.13

Same table, normalized by TMM

Gene	Control	Treatment	log2(T/C)
Gene1	23570	47140	+1
Gene2	23570	47140	+1
Gene3	47140	47140	0
Gene4	47140	47140	0
Gene5	47140	47140	0
Gene6	47140	47140	0
Gene7	47140	47140	0
Gene8	47140	47140	0
Gene9	47140	47140	0
Gene10	94281	47140	-1
Gene11	235702	942809	+2

Now we can see which genes are *really* changing.

Why wasn't CPM good enough?

- CPM already normalizes for sequencing depth
- But CPM does not account for "compositional bias"
- Because sequencing depth is limited and independent of the biology, genes compete for a limited supply of sequence reads
- If one gene goes up, all others have to go down
- When high-abundance genes change, they can have a drastic effect on all others
- This competition for limited sequencing depth affects the counts of all genes, but has no bearing on the biology, so it requires normalization
- TMM fixes this by assuming that the "average" gene is not changing

Real-world CPM fail: globin reduction

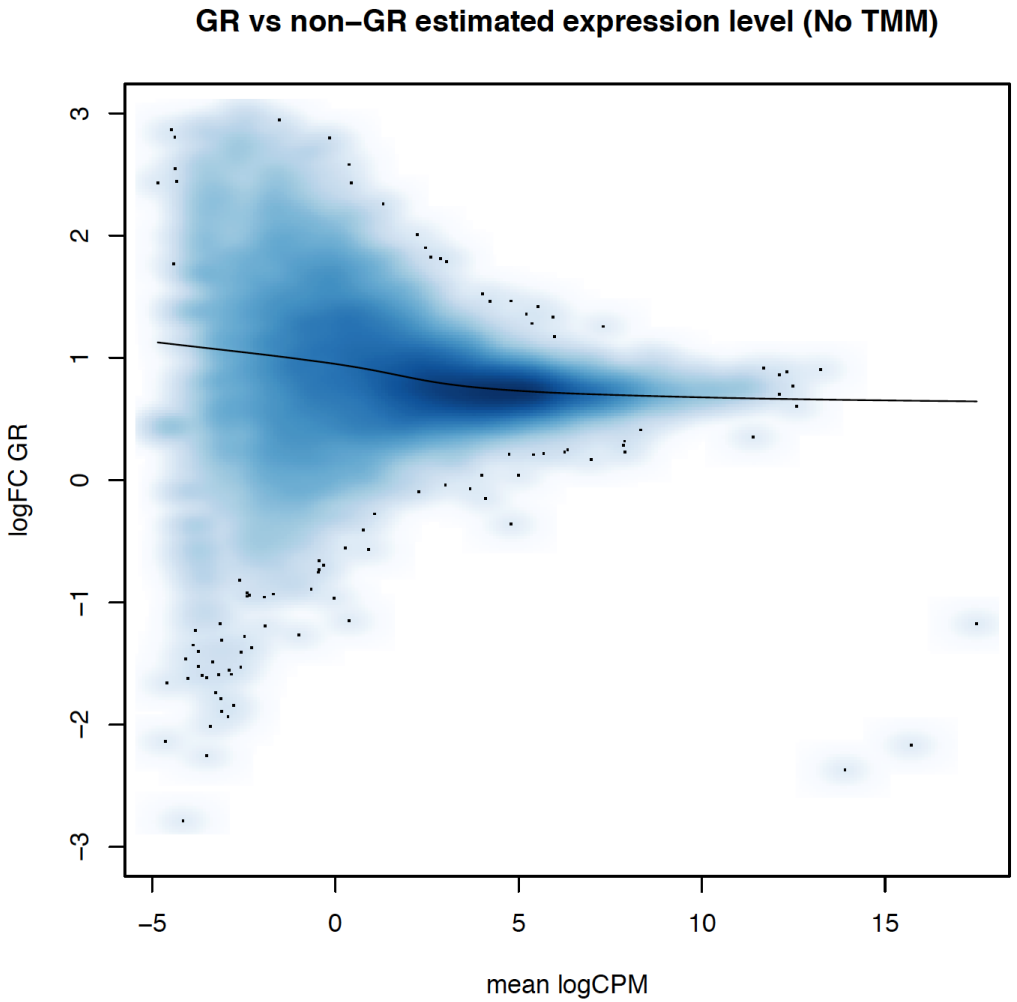

Figure 3: MA plot (Raw logCPM)

Real-world TMM success

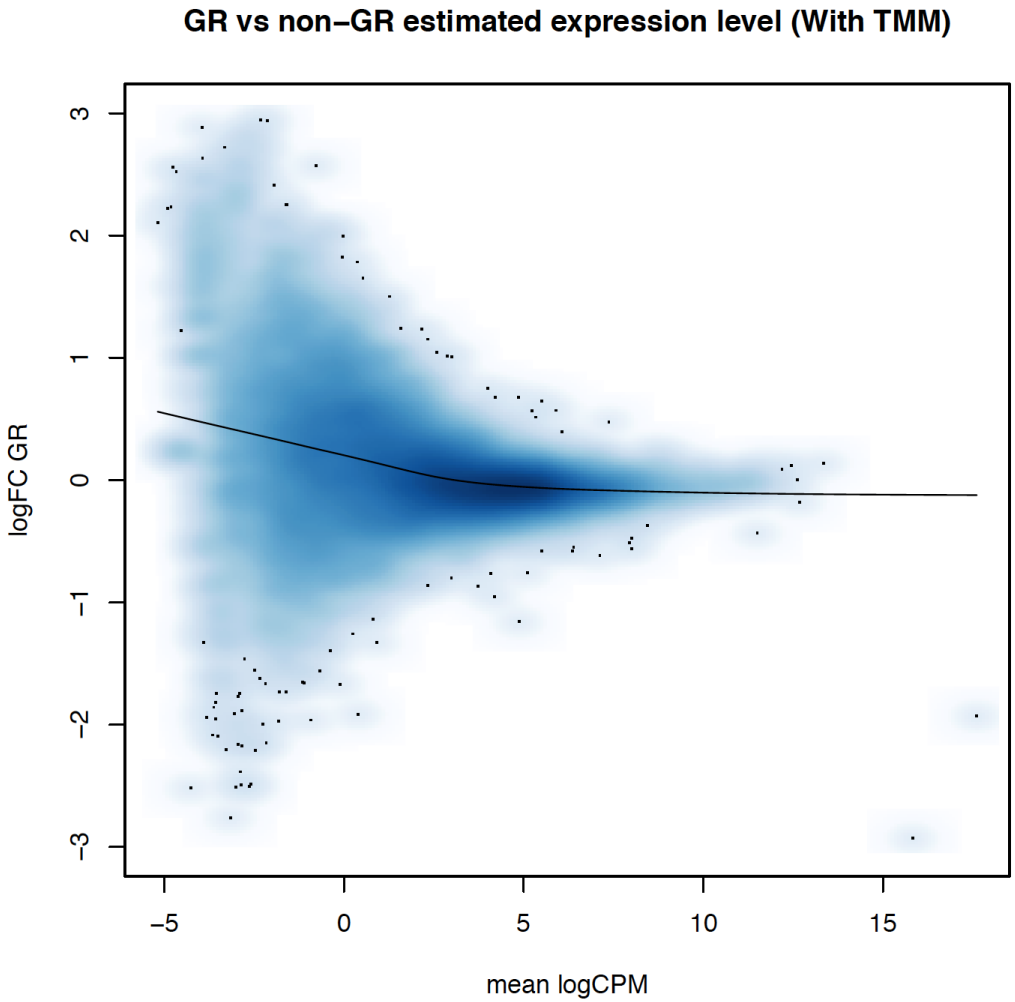

Figure 4: MA plot (With TMM)

FPKM: Normalizing for gene length?

- If gene A and gene B both have 10 CPM in a sample, are they expressed at the same level?
- What if gene A's transcript is 1000 nt long while gene B is 100000?
- If we divide CPM by the transcript length, we get the count of "fragments per kilobase per million fragments sequenced", a.k.a. FPKM

FPKM: Normalizing for gene length?

- If gene A and gene B both have 10 CPM in a sample, are they expressed at the same level?
- What if gene A's transcript is 1000 nt long while gene B is 100000?
- If we divide CPM by the transcript length, we get the count of "fragments per kilobase per million fragments sequenced", a.k.a. FPKM

- This can be used to compare across genes, but it is **not** useful for differential expression tests, as we will see later.
- More recently: TPM, which is like FPKM but comparable across samples
- FPKM/TPM still require composition normalization (i.e. TMM or similar)

Better Variance Estimation with limma and voom

Heteroskedasticity: Easier to understand than to say it

- In ideal data, the mean and variance are independent: every measurement has the same precision. This desirable property is called "homoskedastic"
- R's `lm()` assumes homoskedasticity by default
- If the precision of a measurement depends on its mean or on other factors, the data are "heteroskedsatic", and the model would benefit from adjusting for this dependency
- We can do this adjustment by adding in weights: more precise measurements get a higher weight, less precise measurements get a lower weight

Counting precision depends on the count

- Which coin is more trustworthy?
  - **Coin A:** Flipped 10 times, 5 heads & 5 tails
  - **Coin B:** Flipped 100 times, 50 heads & 50 tails

Counting precision depends on the count

- Which coin is more trustworthy?
  - **Coin A:** Flipped 10 times, 5 heads & 5 tails
  - **Coin B:** Flipped 100 times, 50 heads & 50 tails

- Coin B is more trustworthy because it has remained fair over a larger number of flips
- higher counts are more precise

Counting precision depends on the count

- Which coin is more trustworthy?
  - **Coin A:** Flipped 10 times, 5 heads & 5 tails
  - **Coin B:** Flipped 100 times, 50 heads & 50 tails

- Coin B is more trustworthy because it has remained fair over a larger number of flips
- higher counts are more precise

- Genes with higher expression and/or longer transcripts get higher counts, so they can be measured more precisely
- Also works within a single gene: if Gene A is upregulated in the treatment relative to the control, then the counts in the treatment are also more precise than the control counts
- Samples with higher sequencing depth are more precise for all genes

Voom: modeling the mean-variance trend

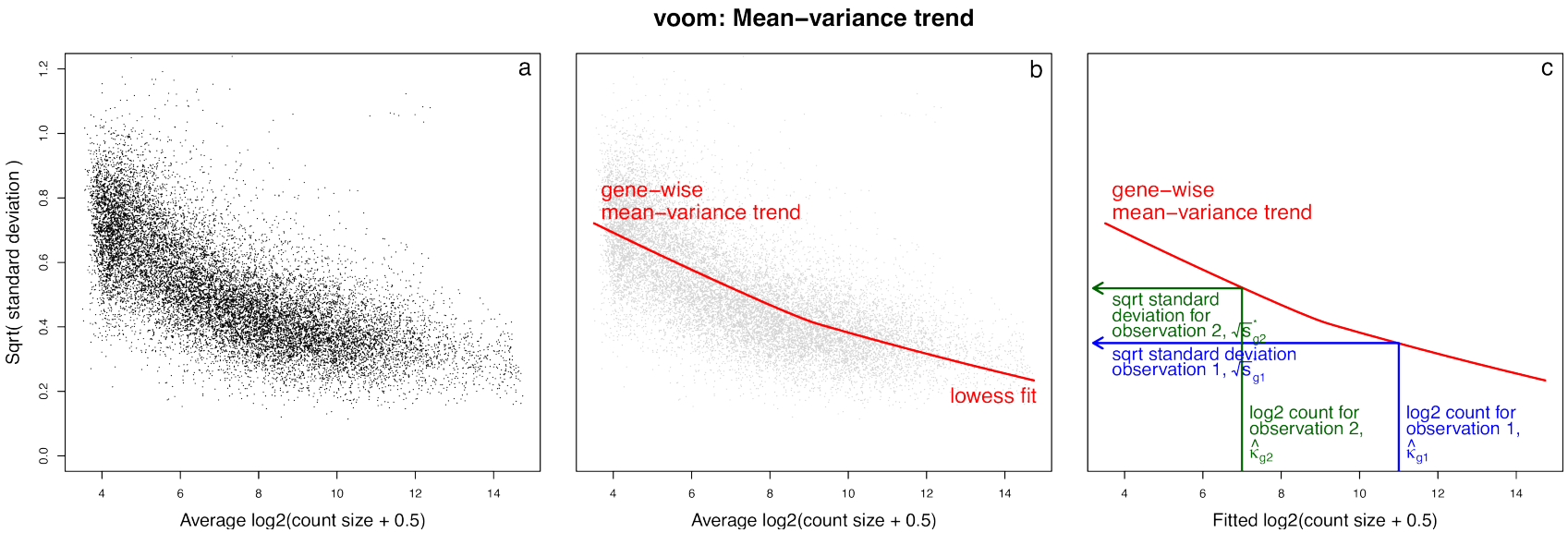

Figure 5: Diagram of voom method

Voom: modeling the mean-variance trend

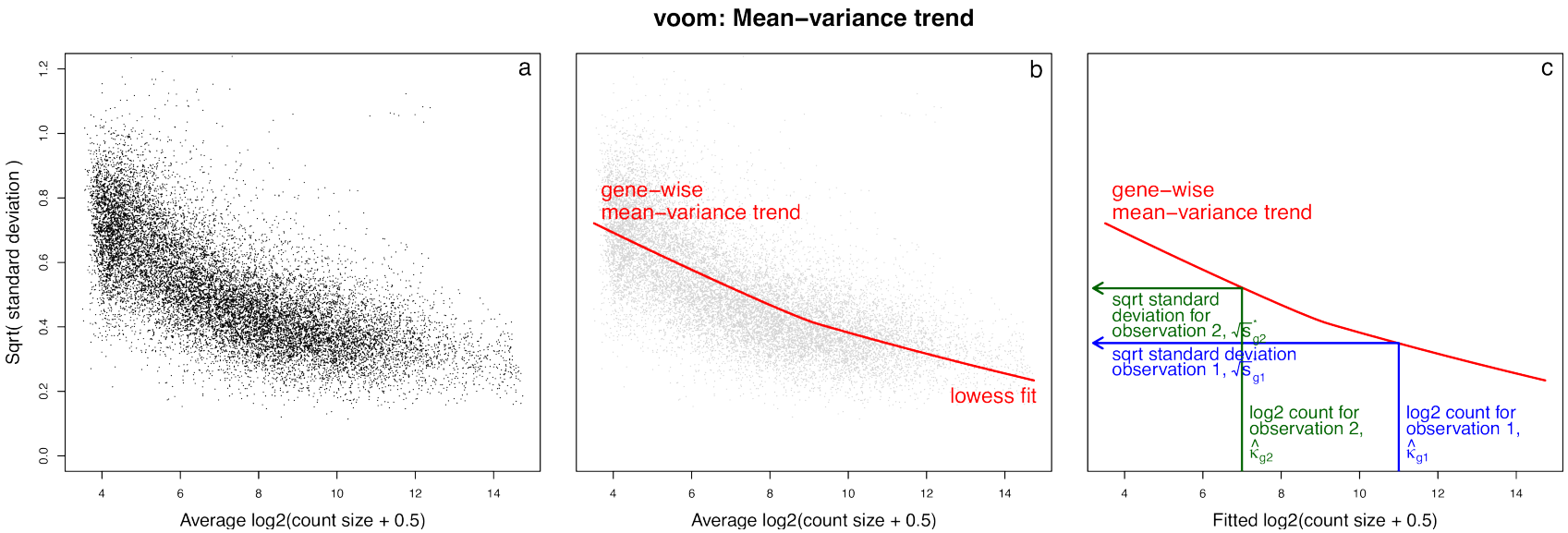

Figure 5: Diagram of voom method

- Lower counts get lower weights, higher counts get higher weights
- Samples with higher depth have higher counts and hence higher weights for all genes on overage

Problem: Few replicates makes it hard to estimate variance

- RNA-seq is expensive, so most experiments have very few replicates
- Few replicates means that we can't get a robust estimate of the variance for each gene
- In turn, the means our p-values are less reliable
- But with 1000s of genes, we *can* get a robust estimate of the average variance of all genes
- This would be great if every gene had the same variance, but we know this isn't the case
- Can we compromise between these two extremes?

Empirical Bayes: Sharing (information) is caring

- We will come up with a scheme where genes partially share information with each other about the variance

1. Estimate variance for each gene normally
2. Take the average of all the genes' variances
3. Set each individual gene's variance somewhere between the gene-specific value and the global average

Empirical Bayes: Sharing (information) is caring

- We will come up with a scheme where genes partially share information with each other about the variance

1. Estimate variance for each gene normally
2. Take the average of all the genes' variances
3. Set each individual gene's variance somewhere between the gene-specific value and the global average

- This result is more accurate than the global average variance *and* more precise than the gene-specific variance
- Empirically better performance in simulations and real tests with positive controls
- Adaptive: as we add more samples, we rely less on the average and more on the gene-specific variances

Empirical Bayes: Sharing (information) is caring

- We will come up with a scheme where genes partially share information with each other about the variance

1. Estimate variance for each gene normally
2. Take the average of all the genes' variances
3. Set each individual gene's variance somewhere between the gene-specific value and the global average

- This result is more accurate than the global average variance *and* more precise than the gene-specific variance
- Empirically better performance in simulations and real tests with positive controls
- Adaptive: as we add more samples, we rely less on the average and more on the gene-specific variances

- Easier to understand visually

Variance estimation: no squeezing

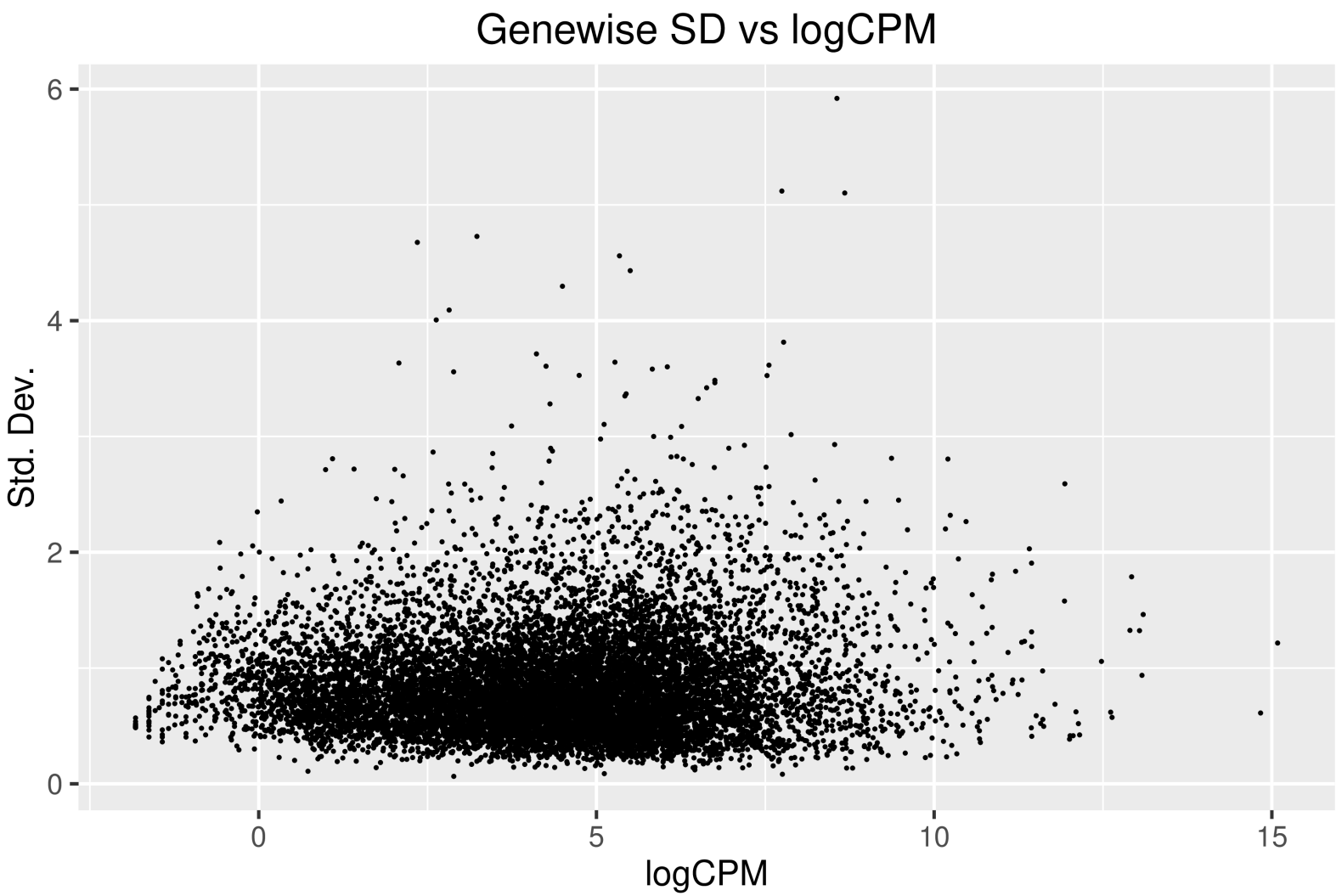

Figure 6: Raw gene-specific SD

Variance estimation: Global average (prior) SD

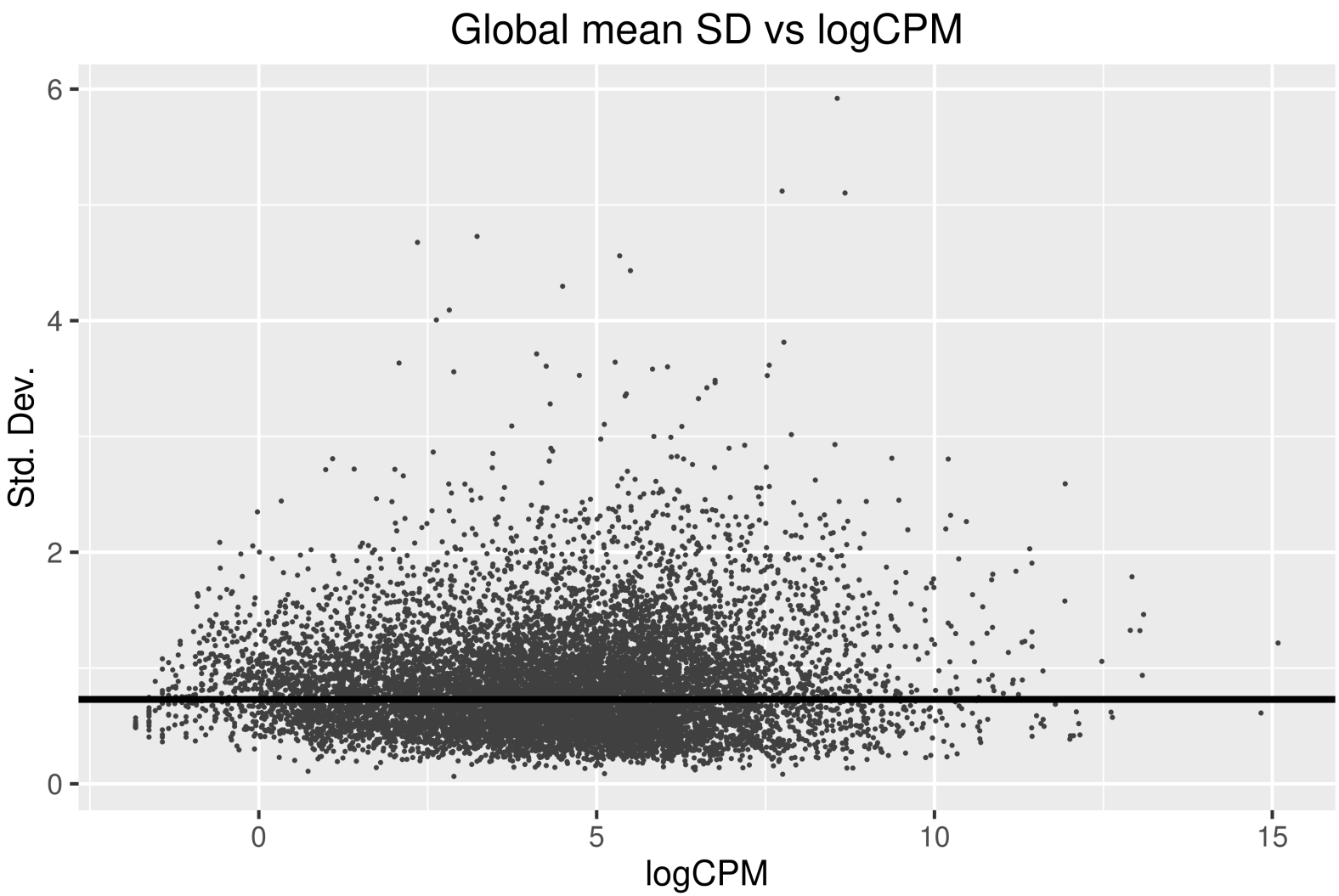

Figure 7: Mean of gene-specific SD

Variance estimation: empirical Bayes squeezing

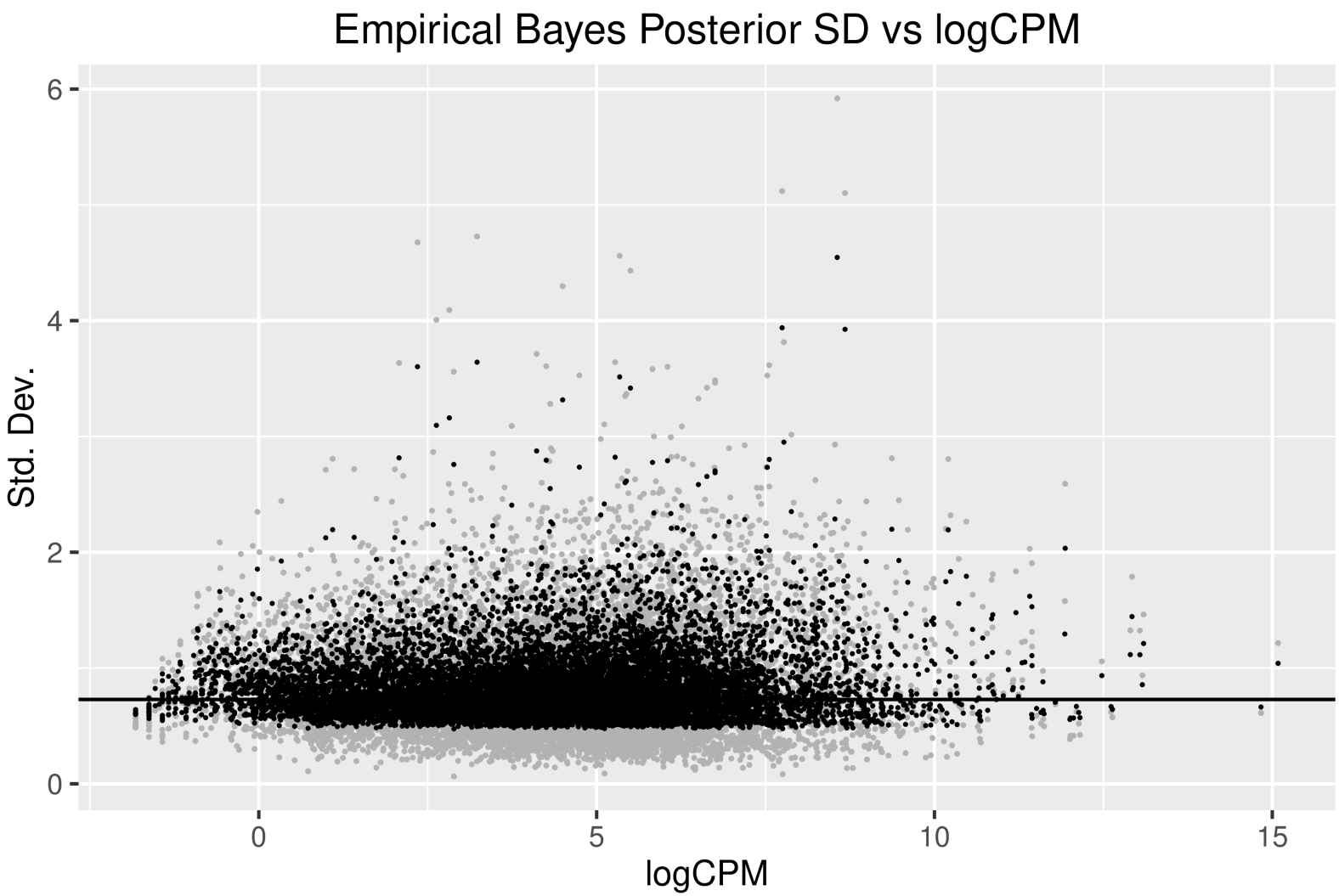

Figure 8: Empirical Bayes squeezed vs raw variances

Multiple Testing Correction; or, why p-values are terrible

Assessing your model with p-value histograms

- By definition, p-values are uniformly distributed under the null hypothesis
- So any deviation from uniformity in multiple tests can be interpreted as deviation from the null hypothesis
- *Technically*, we're not doing multiple tests, but we *are* testing multiple genes, and that's close enough to exploit the aggregate properties of p-values

FDR: Important definitions and distinctions

- FDR: expected number of false positives in a *list* of genes – does not tell the probability of any one gene being a false positive
- Important: FDR is a general term for any false discovery rate calculation – remember to specify the specific method of computing FDR in your Methods section
- Benjamini-Hochberg: an FDR algorithm; puts an *upper bound* on the FDR
- π_0: Estimated proportion of all null hypotheses that are true (non-DE genes), a.k.a. prior probability of non-DE
- q-value: Another commonly used algorithm for estimation of FDR; more liberal than BH, but has a chance to overestimate significance
    - Specifically: q-value equals BH FDR times π_0; or equivalently BH FDR is q-value under the pessimistic assumption that $\pi_0 = 1$

FDR: Important definitions and distinctions

- FDR: expected number of false positives in a *list* of genes – does not tell the probability of any one gene being a false positive
- Important: FDR is a general term for any false discovery rate calculation – remember to specify the specific method of computing FDR in your Methods section
- Benjamini-Hochberg: an FDR algorithm; puts an *upper bound* on the FDR
- π_0: Estimated proportion of all null hypotheses that are true (non-DE genes), a.k.a. prior probability of non-DE
- q-value: Another commonly used algorithm for estimation of FDR; more liberal than BH, but has a chance to overestimate significance
  - Specifically: q-value equals BH FDR times π_0; or equivalently BH FDR is q-value under the pessimistic assumption that $\pi_0 = 1$
- These definitions are best understood in graphical terms

Typical P-value distribution: all null

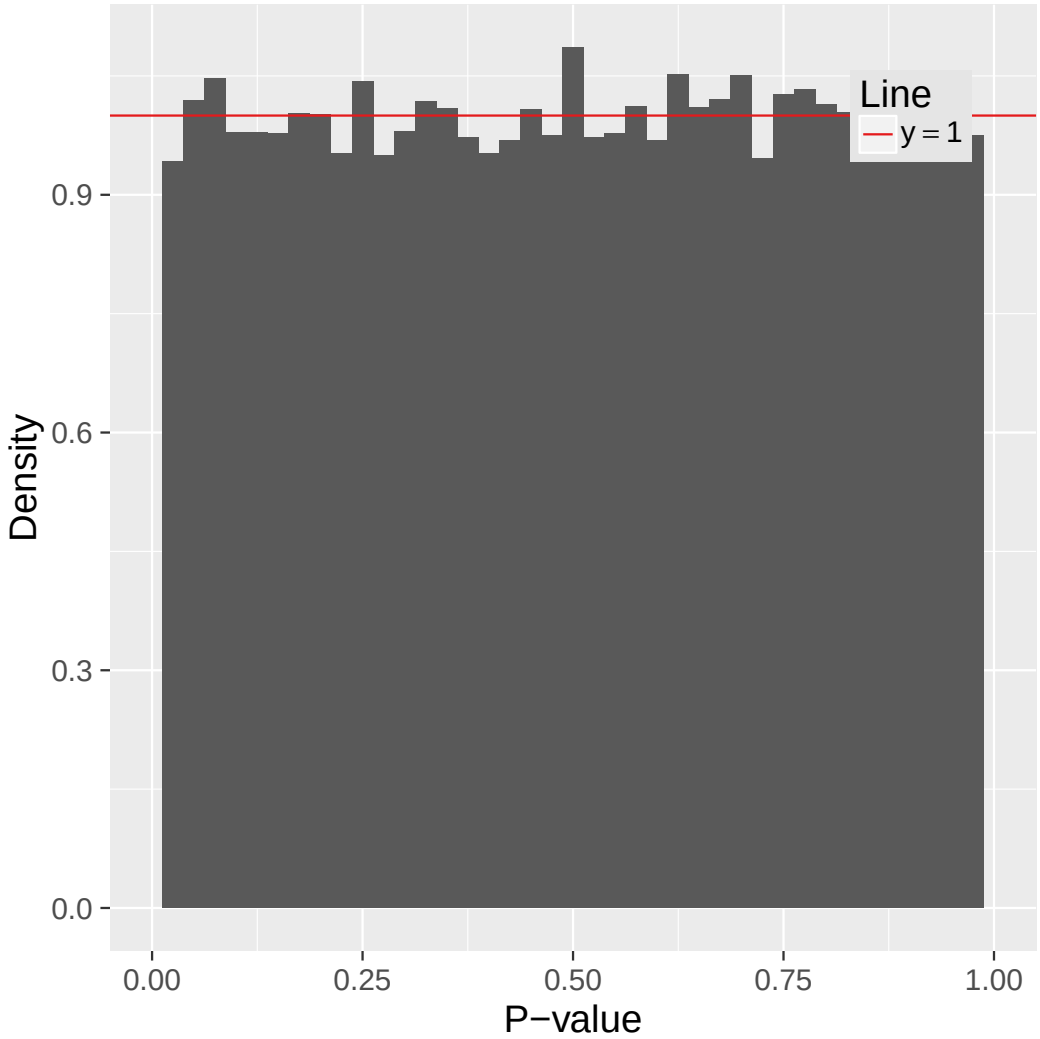

Figure 9: P-value distribution with no signal

Typical P-value distribution: moderate signal

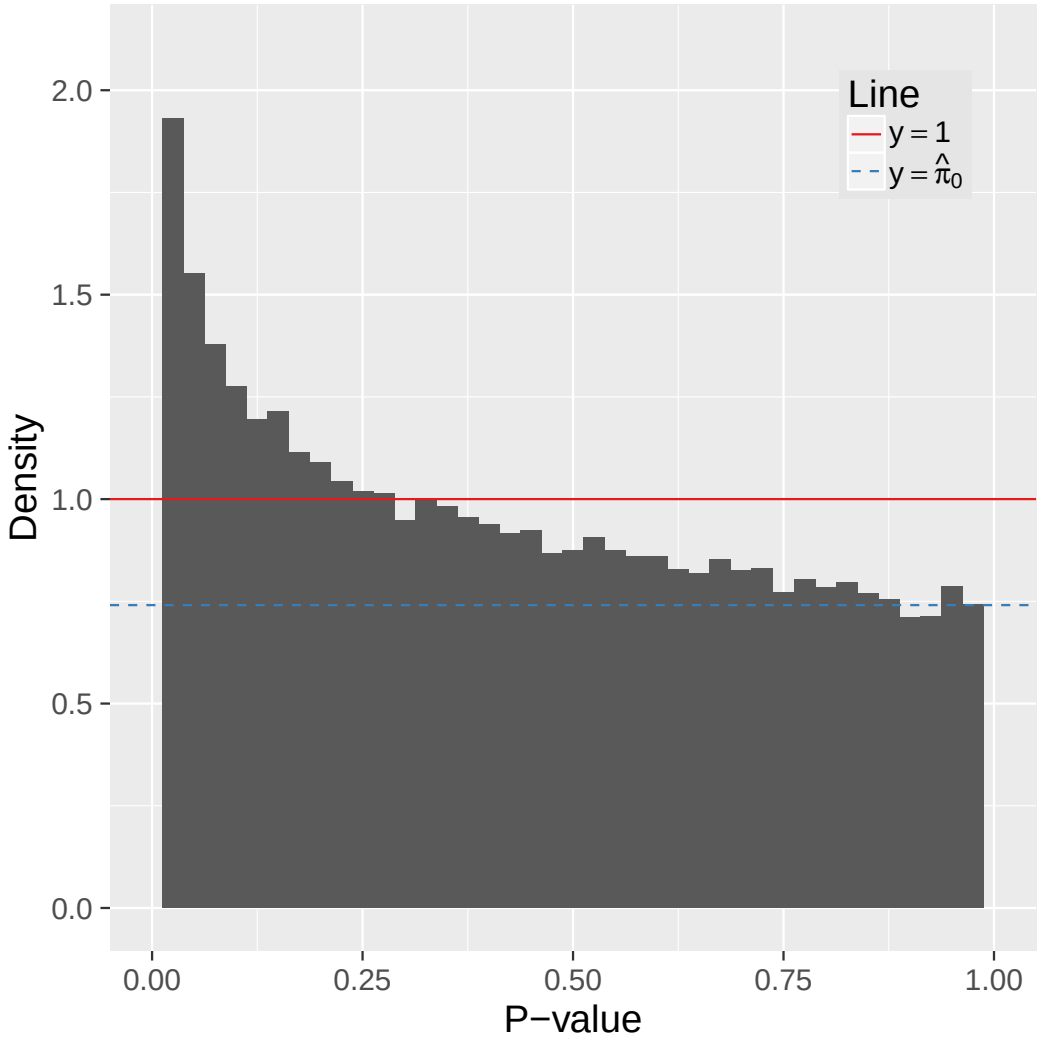

Figure 10: P-value distribution with moderate signal

Typical P-value distribution: moderate signal

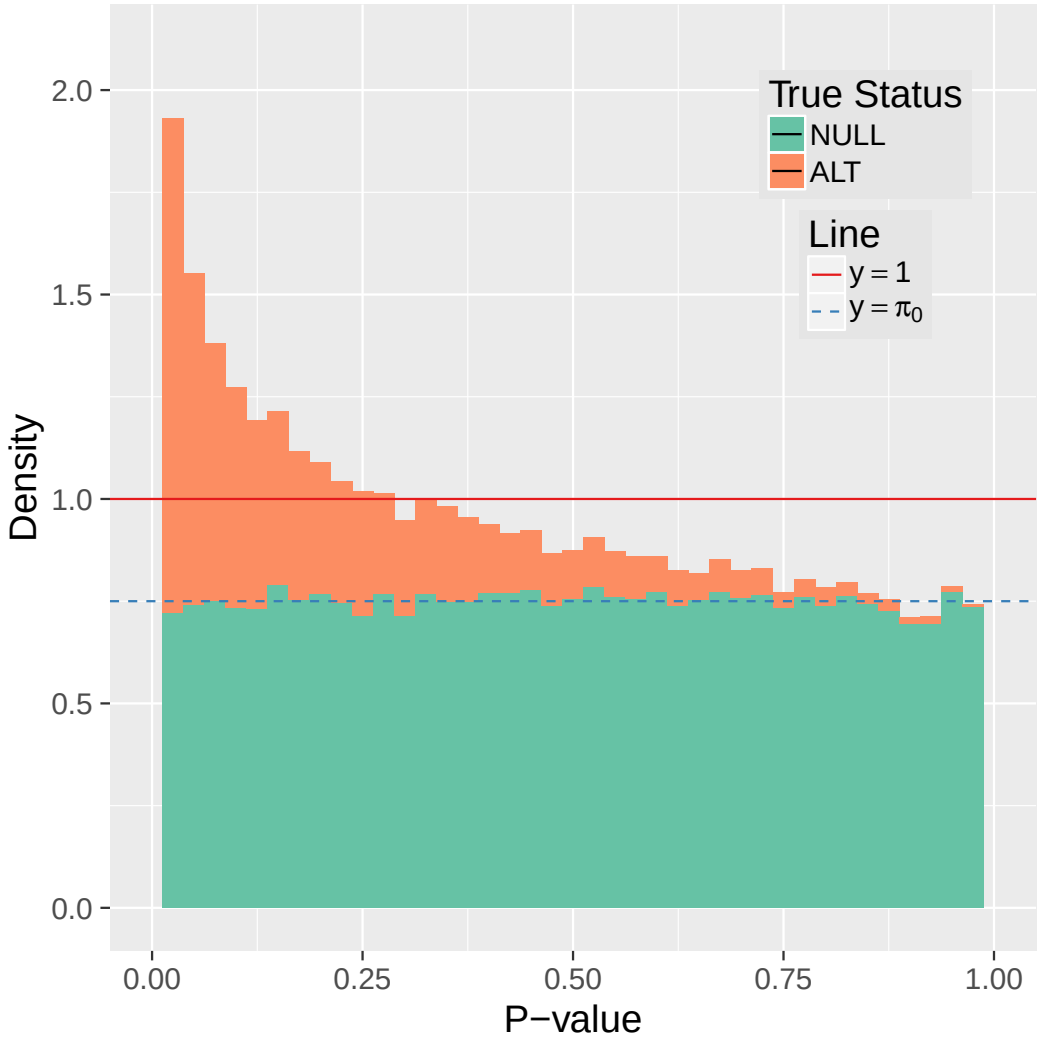

Figure 11: P-value distribution with moderate signal, colored by true status

Evaluating your model using the p-value distribution

- Every p-value distribution should either be uniform or zero-biased
- Any other distribution indicates that your model does not fit the data - Go back and fix your model!
- FDR methods will not have a useful interpretation for an invalid p-value distribution
- Possible causes:
  - Critical assumptions of your model are severely violated (e.g. heteroskedasticity, wrong distribution, excess outliers)
  - Important covariates/batch effects not included in your model
  - Highly correlated covariates are splitting the effect size
  - Unobserved batch effect or other confounding factor is obscuring the signal
  - You accidentally treated continuous variable as categorical or vice versa (common R pitfall!)
- CANNOT be explained by simple lack of signal or excess noise

Atypical P-value distribution: Over-Conservative

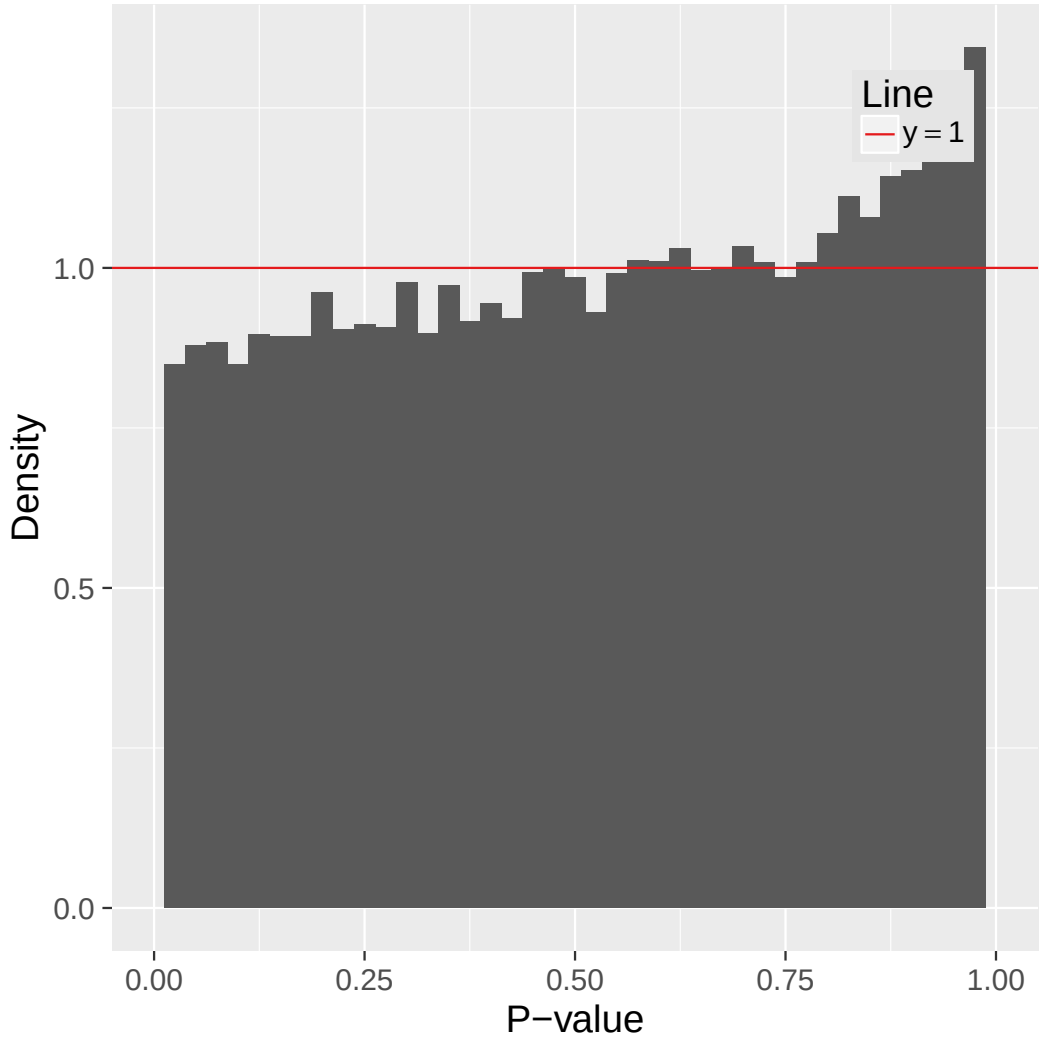

Figure 12: P-value distribution, worse than uniform

Atypical P-value distribution: Bimodal

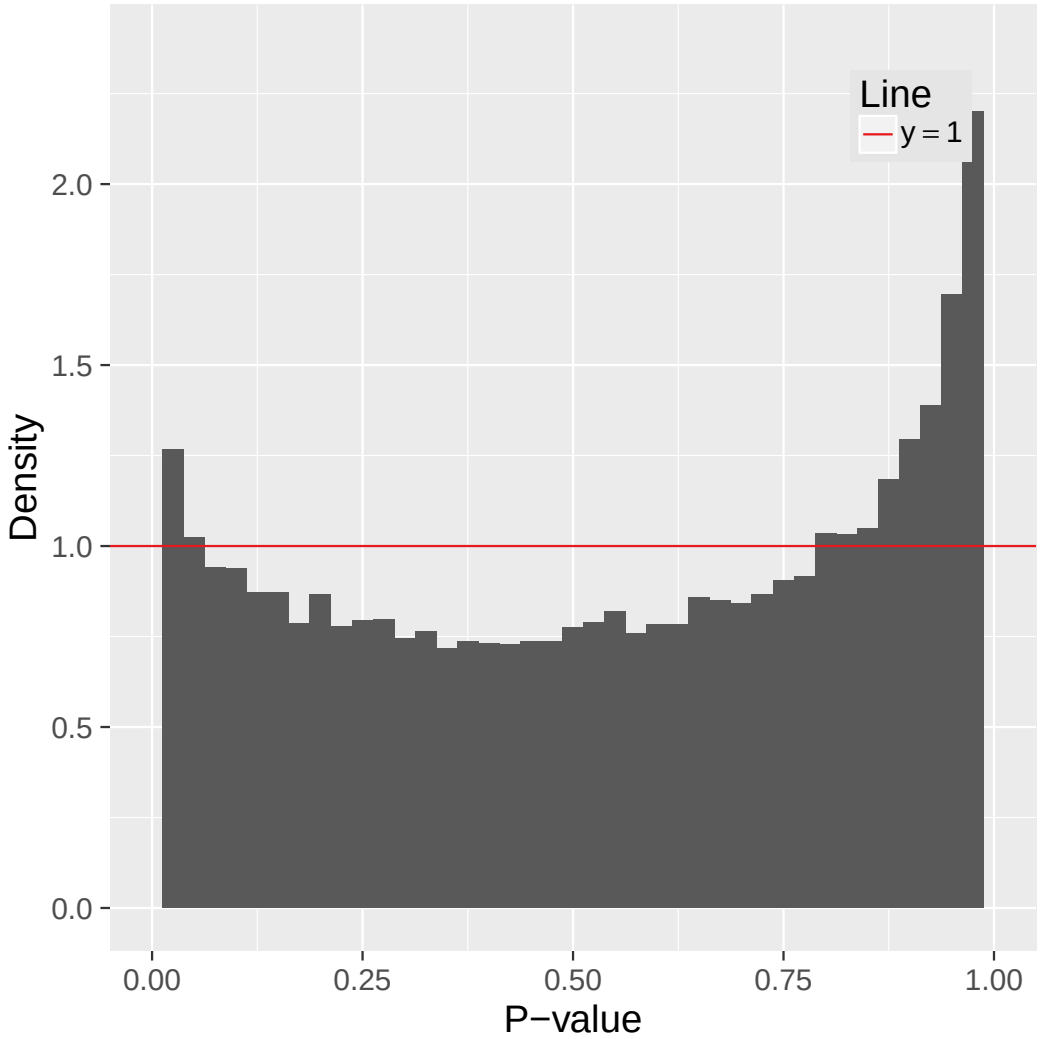

Figure 13: P-value distribution, bimodal

Atypical P-value distribution: Bump in the middle

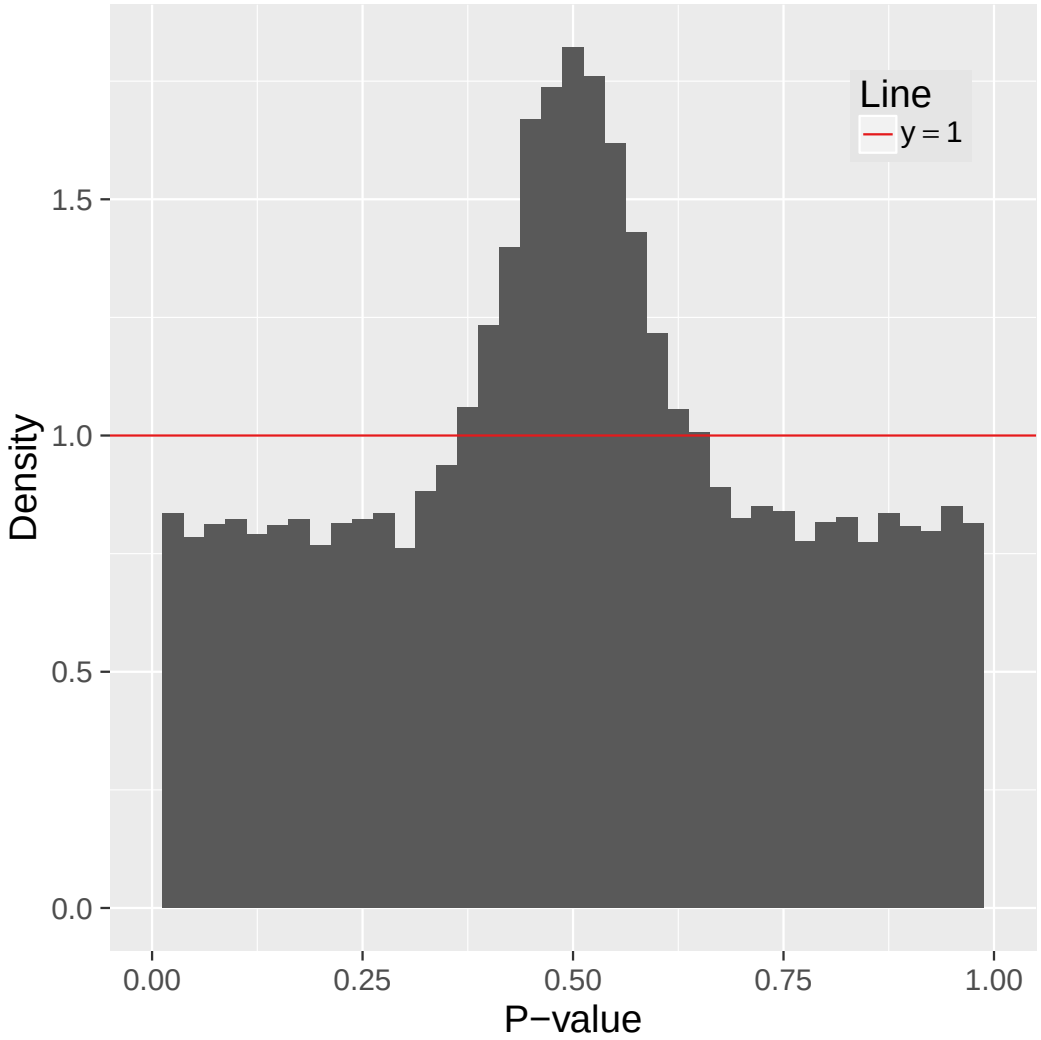

Figure 14: P-value distribution, non-monotonic

Atypical P-value distribution: Discrete values

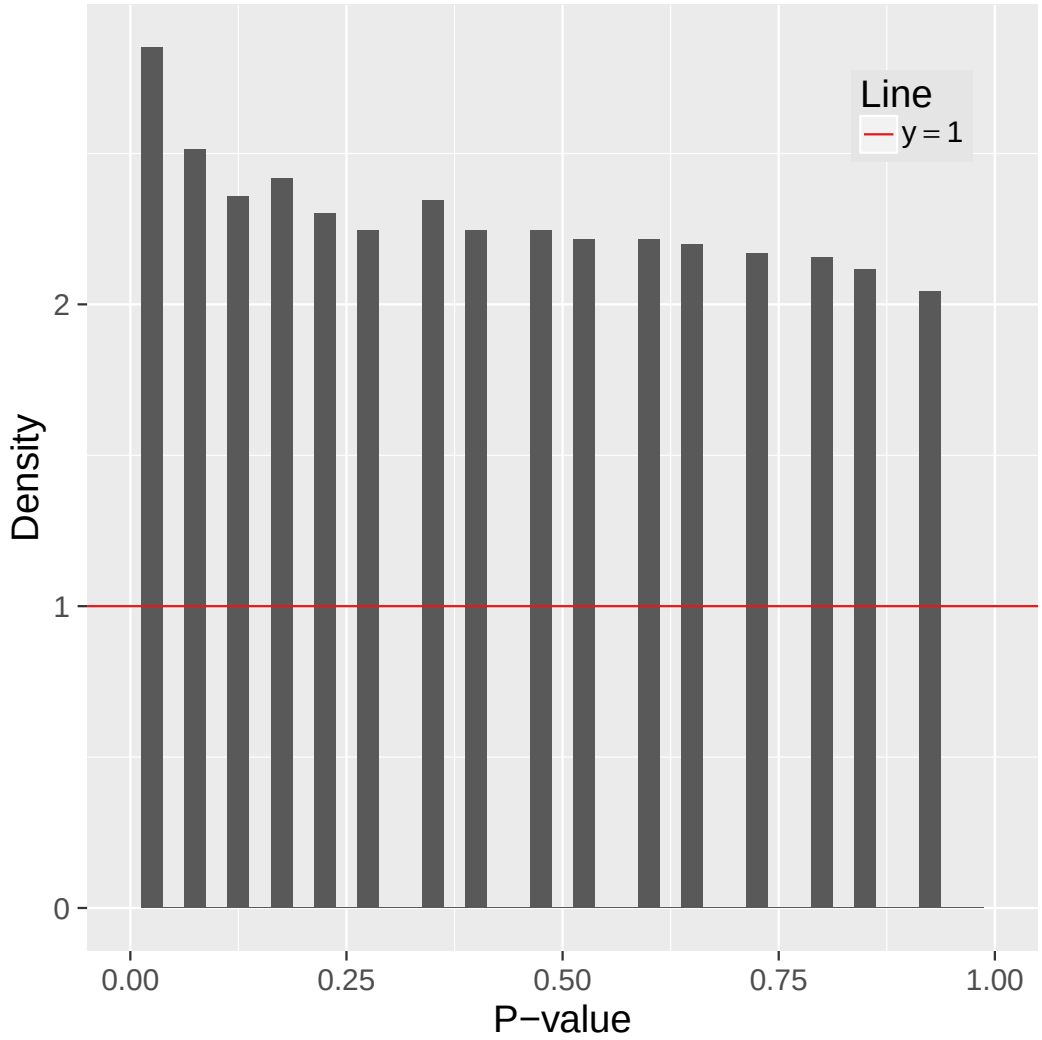

Figure 15: P-value distribution, discrete

Conclusions

- RNA-seq read are aligned and counted to obtain counts for each gene in each sample
- Counts are log-transformed and analyzed using an ordinary linear model...

Conclusions

- RNA-seq read are aligned and counted to obtain counts for each gene in each sample
- Counts are log-transformed and analyzed using an ordinary linear model...

- ...with modifications to account for:
- normalization (TMM)
- counting precision (voom)
- variance estimation (empirical Bayes squeezing/information sharing)
- Luckily, the limma package does does all of this extra work for you, so it's almost as easy as normal `lm()`
- Finally, p-values are adjusted for multiple testing to obtain FDRs

Not Pictured: All of these other things

There's a lot more that you can do with RNA-seq data!

- Differential expression using `glm` and the negative binomial distribution
- Estimation of alternative isoform expression levels
- Differential splicing analysis
- Uneven coverage across gene bodies
- Gene set/pathway enrichment testing for differentially expressed/spliced genes
- Co-expression networks: WCGNA
- Variant calling of coding SNPs
- Association of SNPs with expression levels (eQTLs)
- Association of TFBS/histone marks/miRNA with expression levels (epigenetics & post-transcriptional regulation)

Not Pictured: All of these other things

There's a lot more that you can do with RNA-seq data!

- Differential expression using `glm` and the negative binomial distribution
- Estimation of alternative isoform expression levels
- Differential splicing analysis
- Uneven coverage across gene bodies
- Gene set/pathway enrichment testing for differentially expressed/spliced genes
- Co-expression networks: WCGNA
- Variant calling of coding SNPs
- Association of SNPs with expression levels (eQTLs)
- Association of TFBS/histone marks/miRNA with expression levels (epigenetics & post-transcriptional regulation)

Many high-throughput technologies are like this. Think carefully about multiple analyses you can do with the same data to get your money's worth!

Any Questions?

Reminder: Hands-on lab session this afternoon

Join us again at 1:00 PM in room 3A-216 for a hands-on session where you can follow along with an analysis of a real, complex data set. Focus will be on the importance of exploratory analysis for selecting the correct model for your data, and what a difference the correct model can make for statistical significance.