

# Read quantification

Physalia - Lecture 2

# Outline

- Counting Mapped Reads
- Intronic/Exonic
- Pseudoaligners
- Gene expression units
- Gene length & coverage
- Isoforms/Splicing
- Counting UMIs

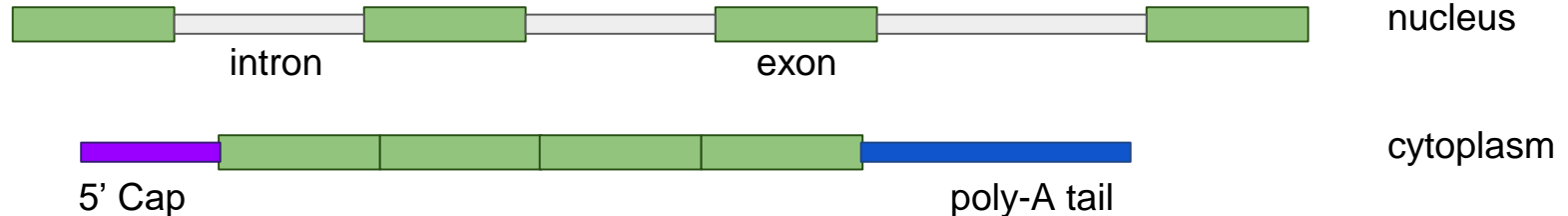
# Counting Mapped Reads

If reads were mapped to the transcriptome:

- Non-exonic regions already excluded
- Each transcript is a separate “chromosome”
- Just need to count reads mapped to each “chromosome”

If reads were mapped to the genome:

- Intronic & exonic regions
- Must overlap mapping locations with annotations

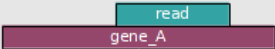
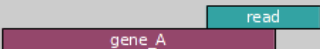
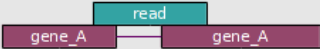


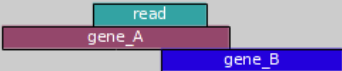
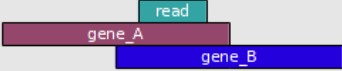



# Counting Mapped Reads

Most popular are HT-Seq & featureCounts (part of Subread)

- These give nearly identical results

Generalizable to any kind of features for which annotations are provided

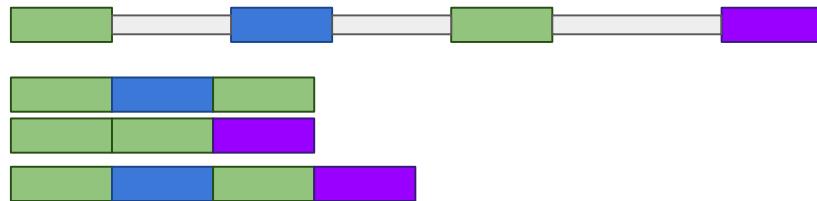
HTSeq options	union	intersection_strict	intersection_nonempty
	gene_A	gene_A	gene_A
	gene_A	no_feature	gene_A
	gene_A	no_feature	gene_A
	gene_A	gene_A	gene_A
	gene_A	gene_A	gene_A
	ambiguous (both genes with --nonunique all)	gene_A	gene_A
	ambiguous (both genes with --nonunique all)		
	alignment_not_unique (both genes with --nonunique all)		

# Counting Reads with Pseudo-Aligners

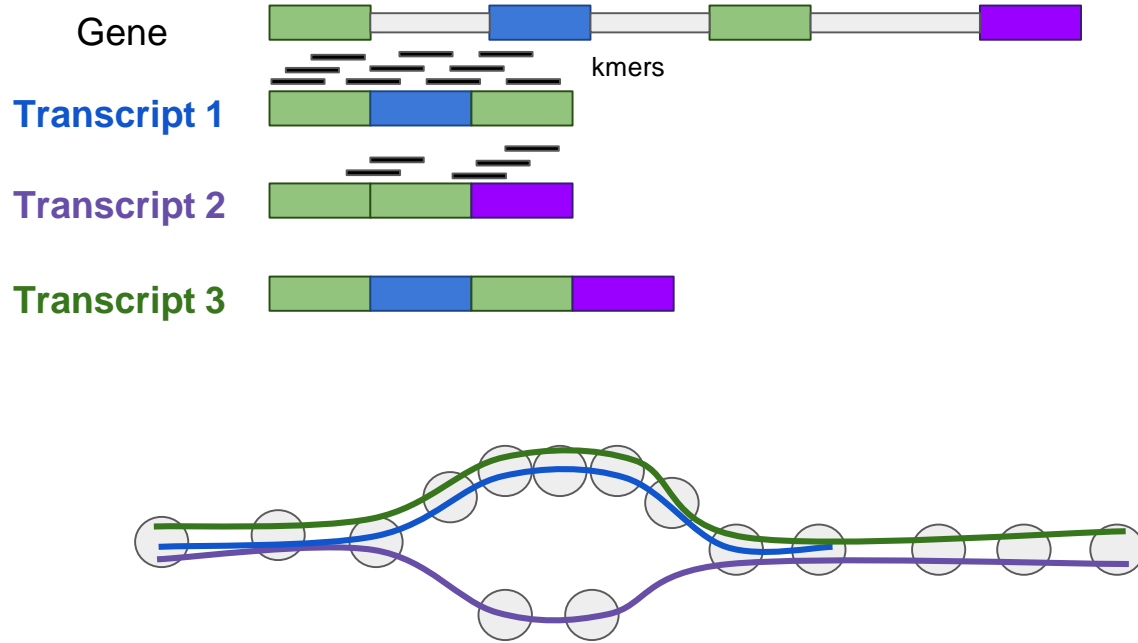
Pseudo alignment breaks reads down into k-mers which are matched to a database of k-mers obtained from the transcriptome. These matches then “vote” on which transcript the read originated from.

Most popular tools: Kallisto/Salmon

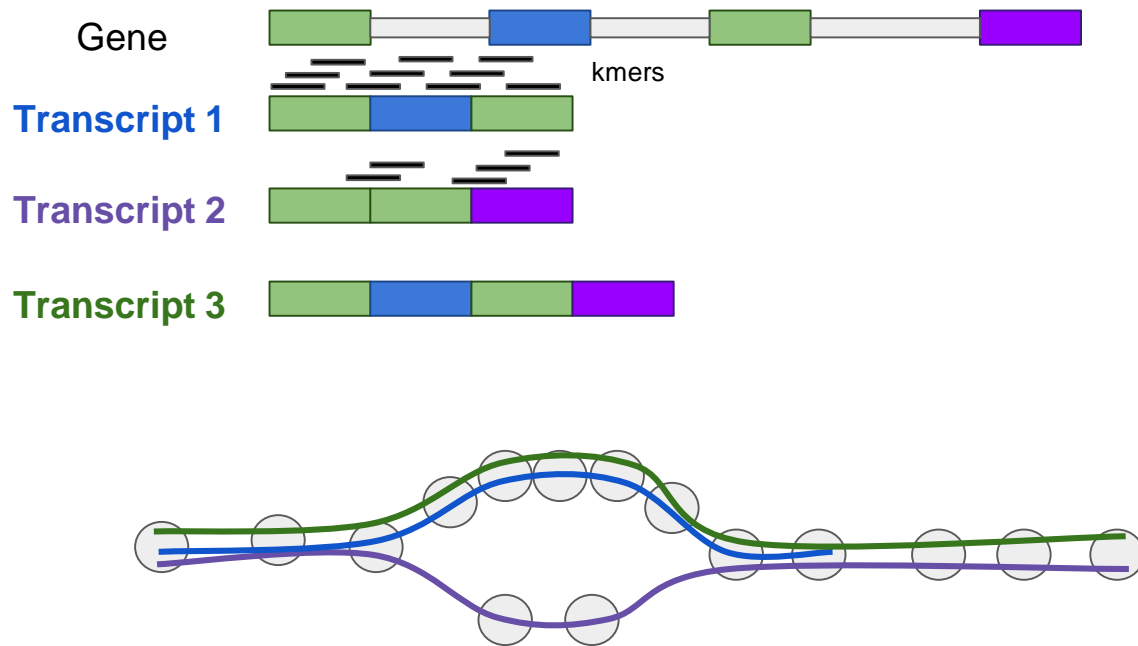
- Cannot align to the genome.
- May generate specific alignments
  - Not as reliable as standard mappers



# Pseudo Aligners : Create a kmer-hash table



# Pseudo Aligners : Create a kmer-hash table



Hash function:  
 $f(\text{k-mer}) = \text{bucket id}$

Buckets of small sets of kmers

B1

$K1 = t1, t2, t3; K8 = t1, t3$

B2

$K3 = t1, t2, t3; K5 = t1, t2, t3$

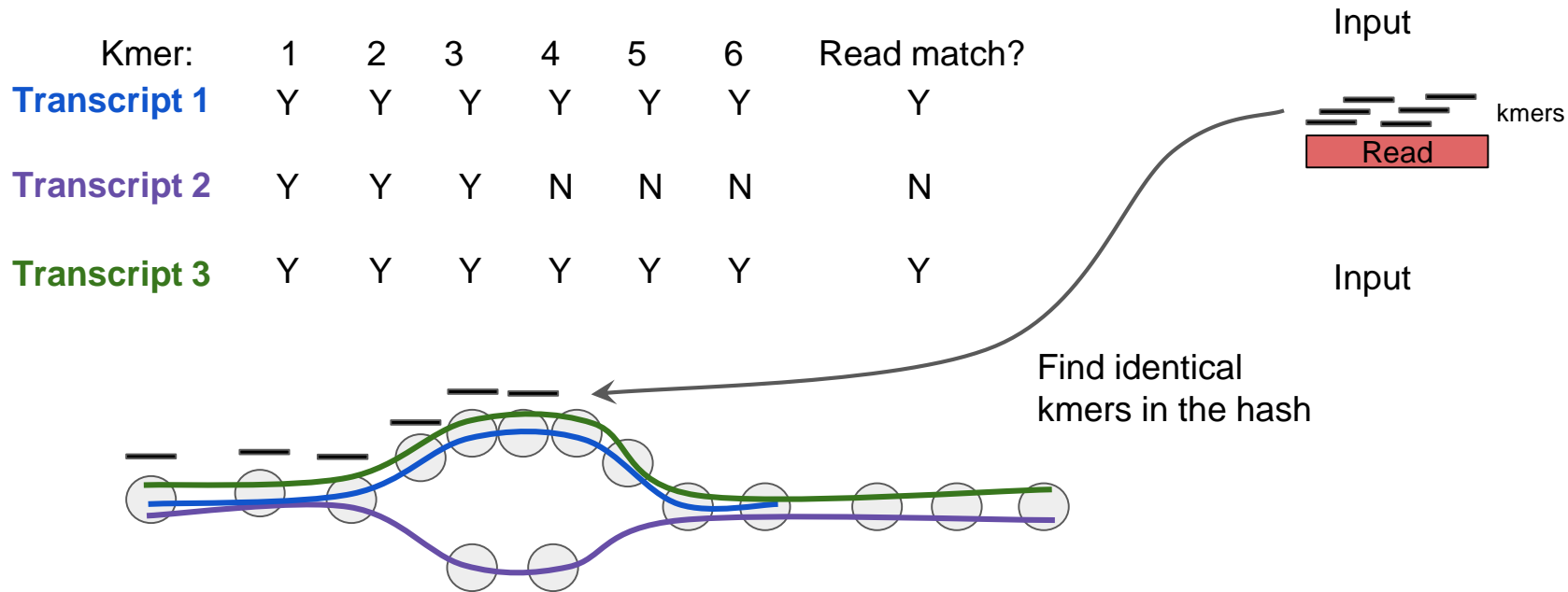
B3

$K4 = t2; K7 = t2, t3$

B4

$K6 = t1, t2; K2 = t2, t3$

# How do Pseudo-Aligners quantify expression?





# How do Pseudo-Aligners quantify expression?

Each read is thus assigned to a set of transcripts, called “equivalence classes”

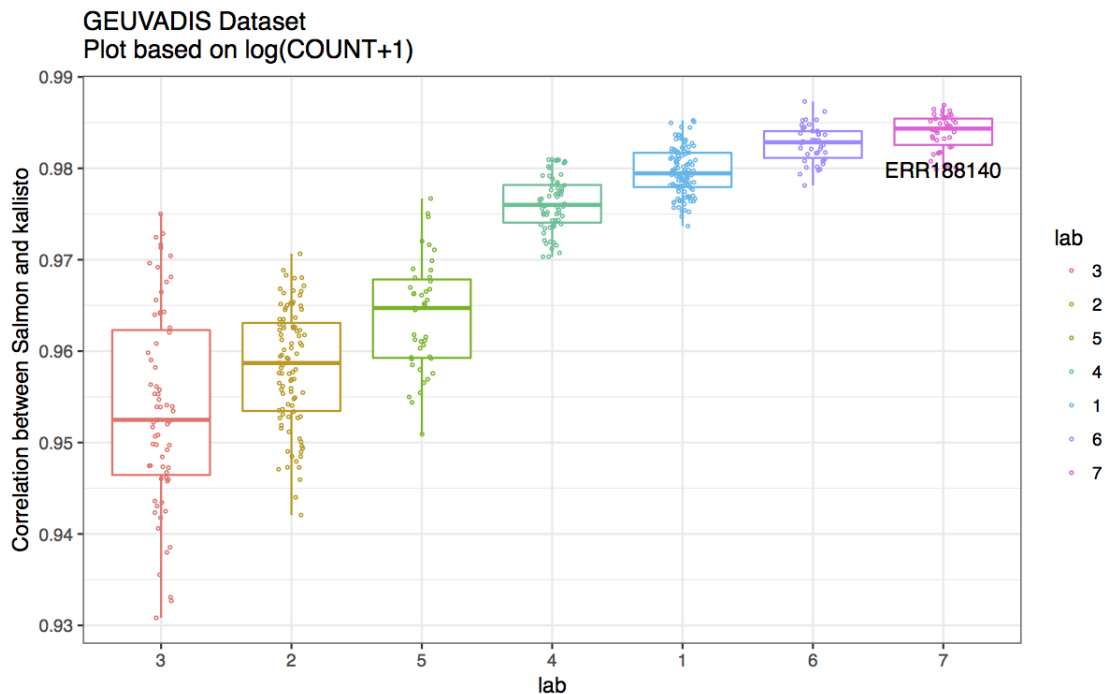
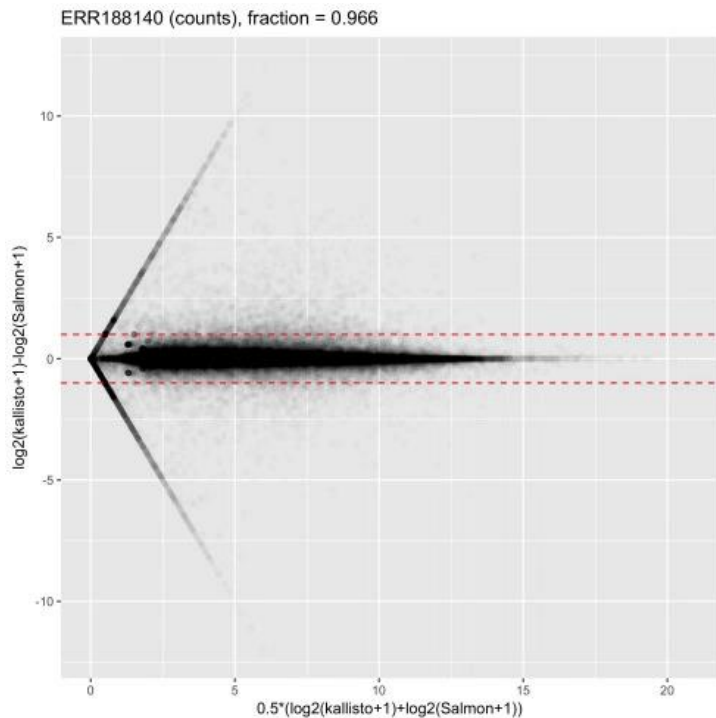
The read counts,  $c_e$ , for each equivalence class are used to estimate transcript level expression by maximum likelihood:

$$L(E_t) = \prod_{t \in e} (\sum E_t / l_t)^{c_e}$$

The “effective length” of a transcript,  $l_t$ , is based on sequence biases from bulk RNASeq, it does not consider the observed transcript coverage.

Transcript counts are estimated by setting  $l_t$  to one. But this does not guarantee integer-valued counts.

# Salmon vs Kallisto



Typical cell-cell correlations from scRNASeq: 0.7-0.9

(left) Pachter, 2017, <https://liorpachter.wordpress.com/2017/09/02/a-rebuttal/>

(right) Patro et al., 2017, <https://github.com/salmonteam/SalmonBlogResponse/blob/master/SalmonBlogResponse.md>

# Counting Reads : Multi-mappers

One challenge with counting reads is how does one count multi-mapping reads?

featureCounts and HT-Seq:

- Discard multi-mapping reads
- Down weight multi-mapping reads

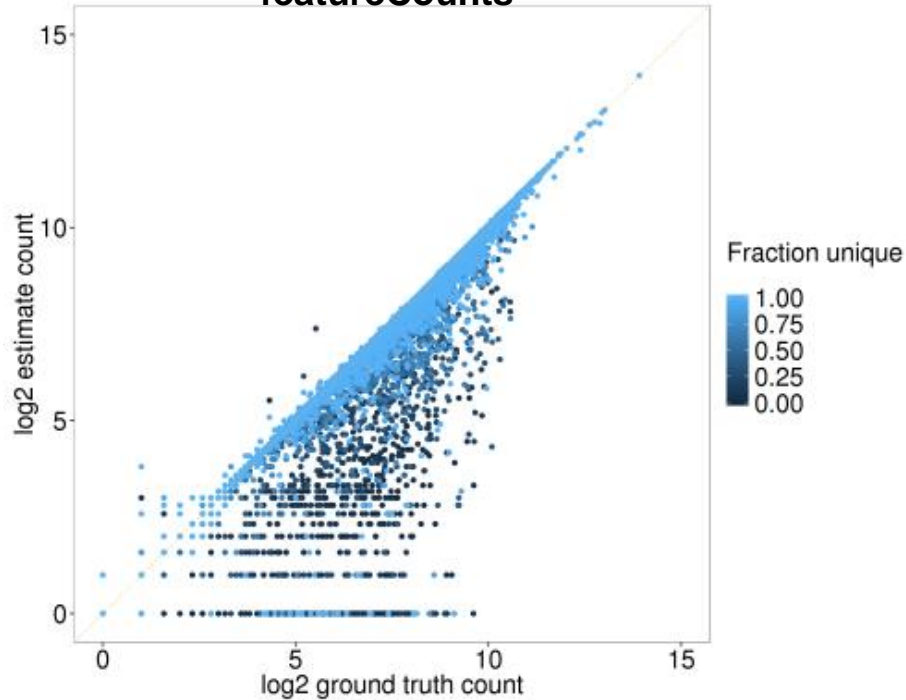
Reasoning:

- 1) Avoid false-positive expression from genes that are not expressed but share homology with expressed genes.
- 2) Prevent multi-mapping reads contributing a disproportionate amount to overall expression levels.

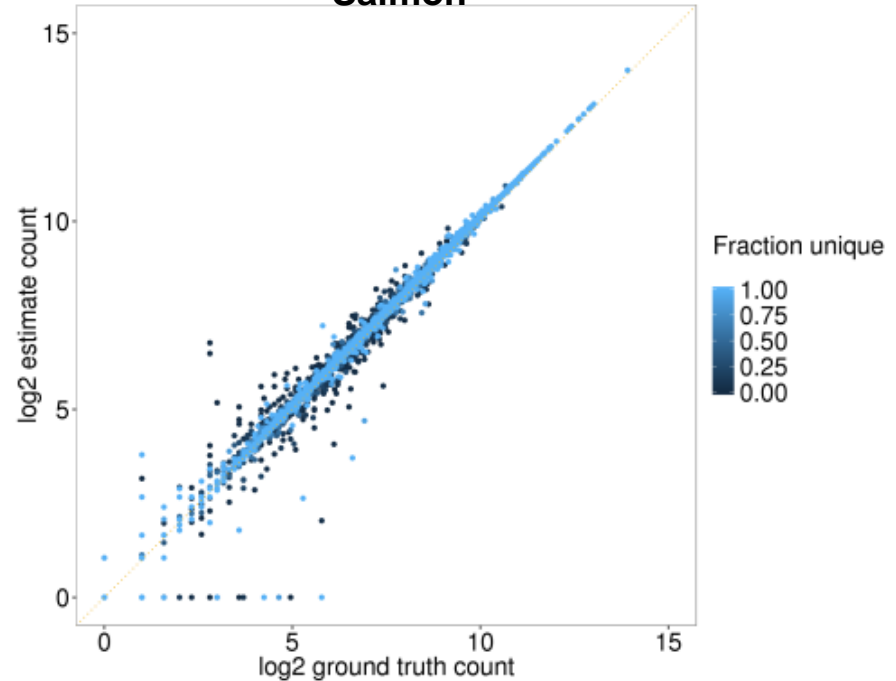
The Flaw: How does this affect the expression estimates of expressed genes with high homology?

# Excluding them biases against homologous genes

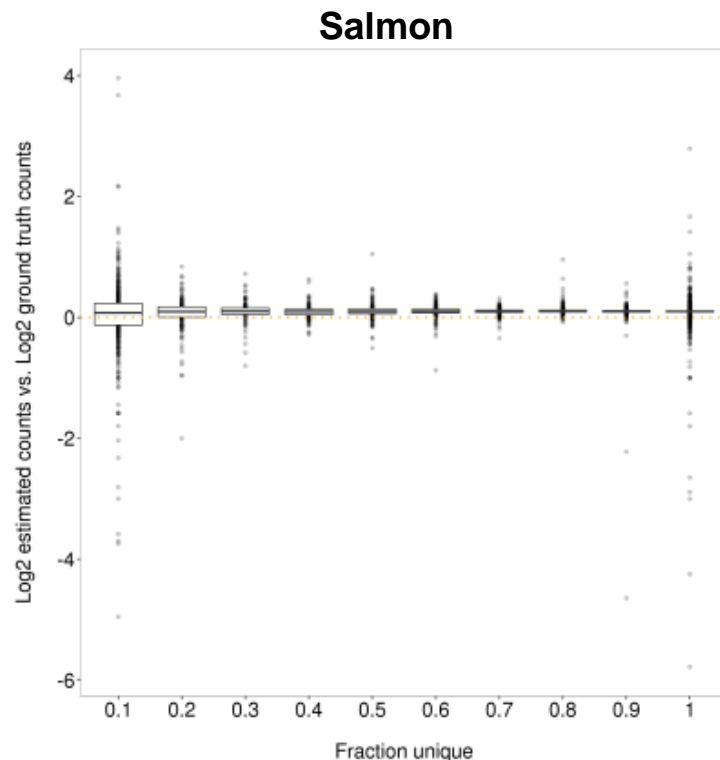
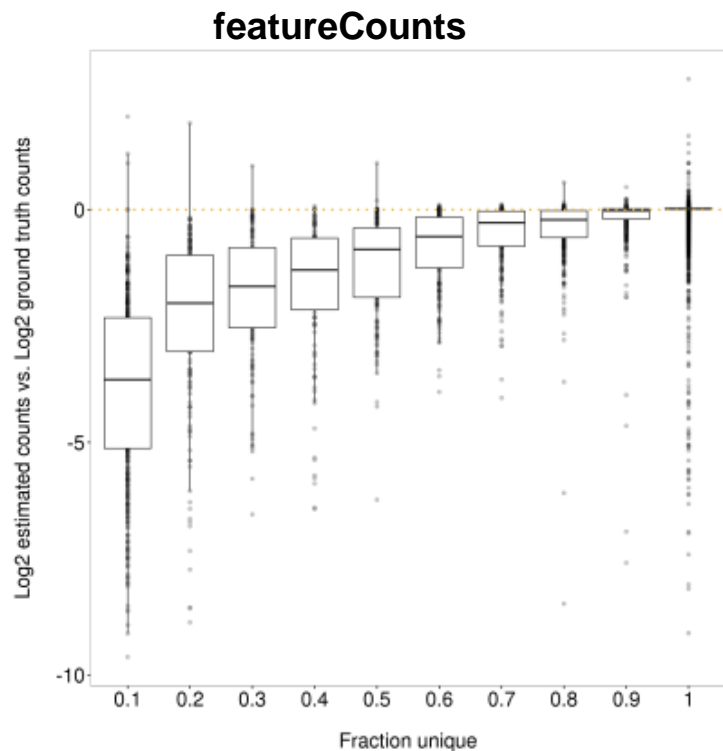
featureCounts



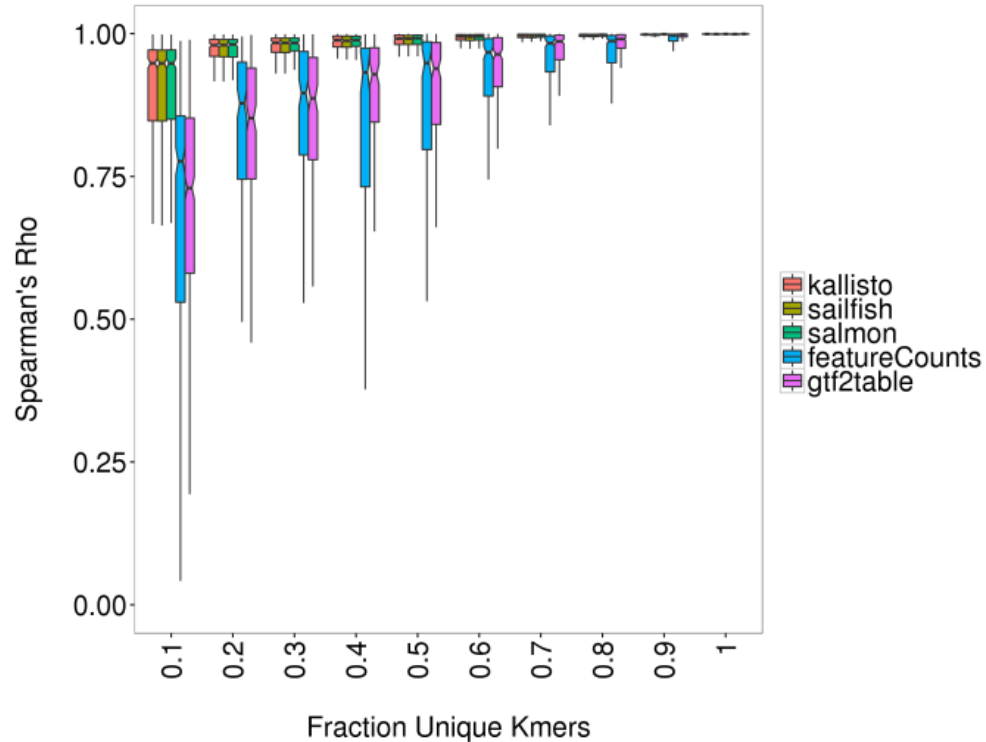
Salmon



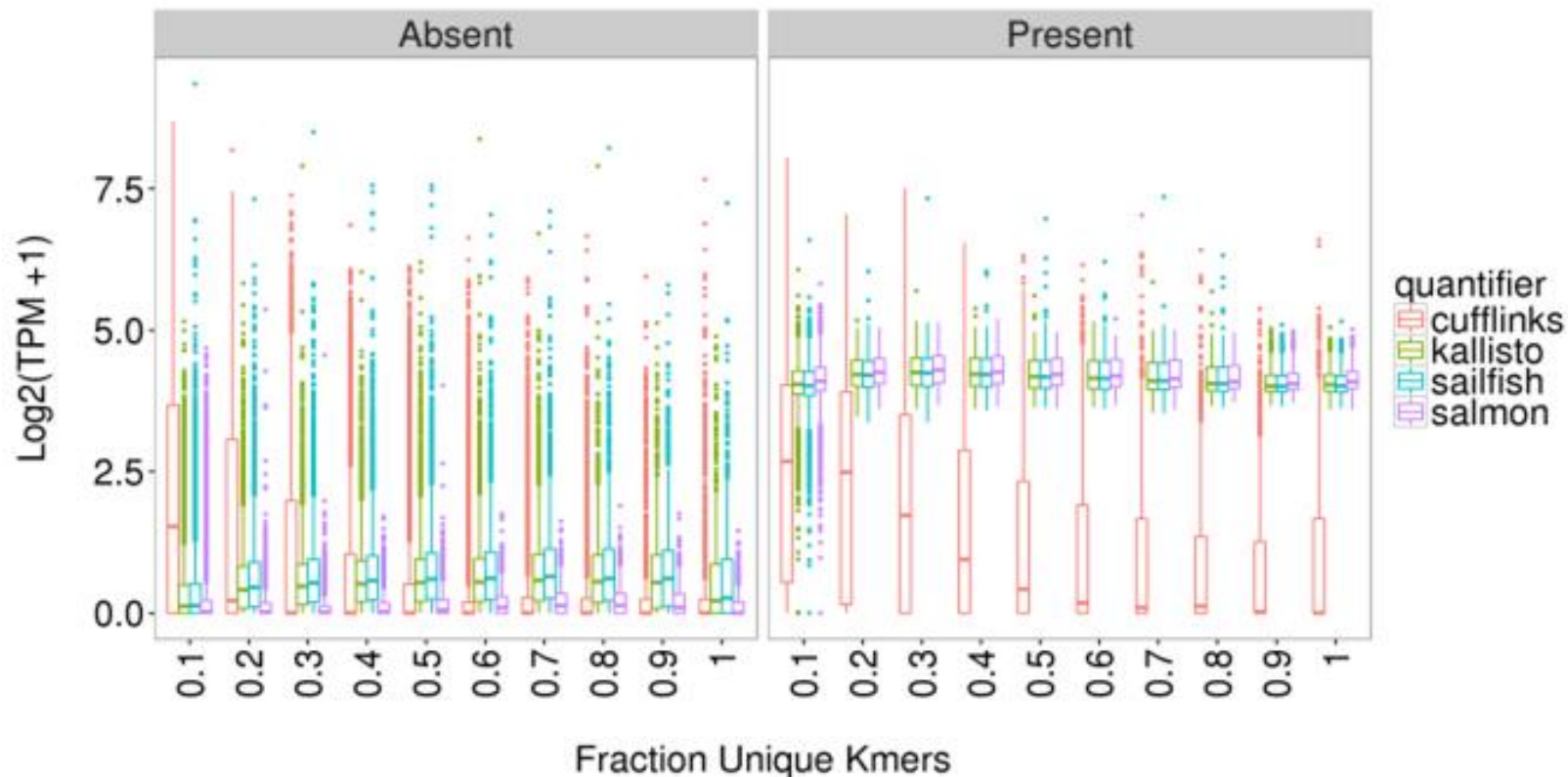
# Excluding them biases against homologous genes



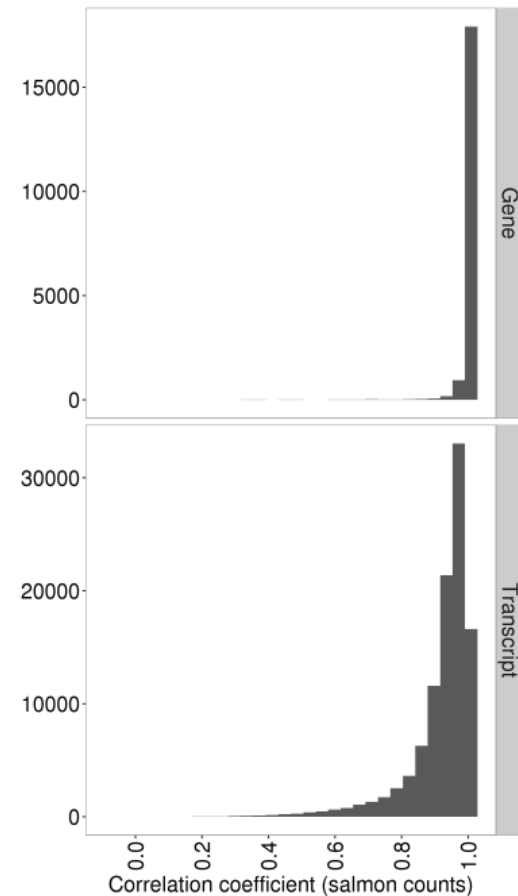
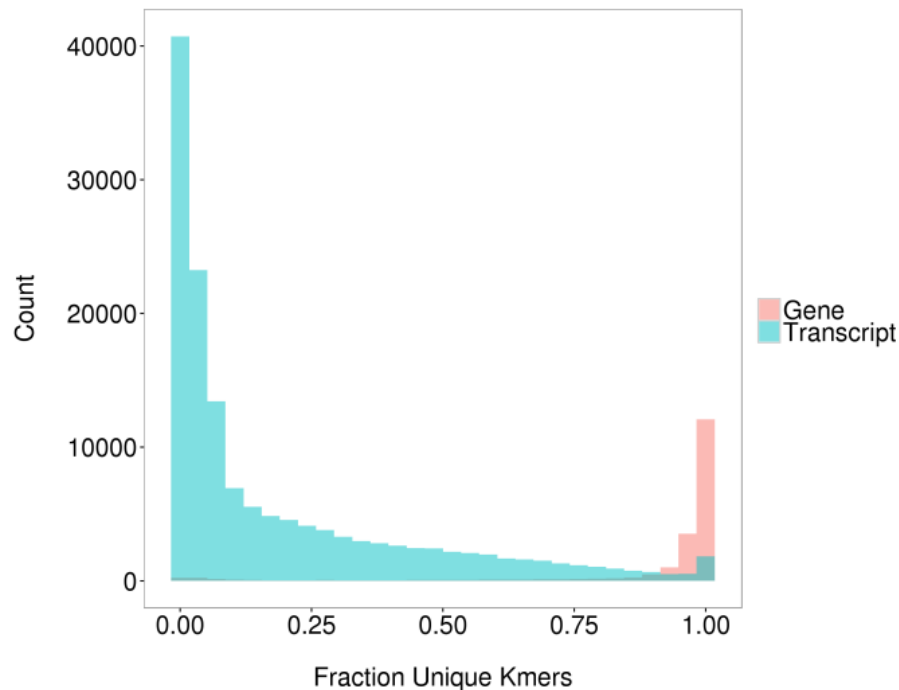
# Does this bias affect cross-sample comparisons?



# But what about unexpressed transcripts?



# Quantification: Genes vs Transcripts





Any Questions?

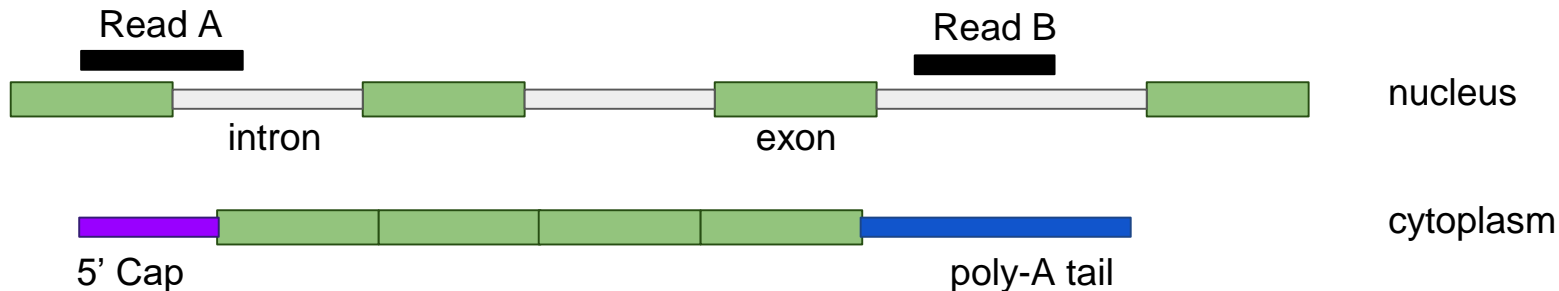
# Discussion: Quantifying Intronic Reads

**Why would you want to quantify intronic reads?**

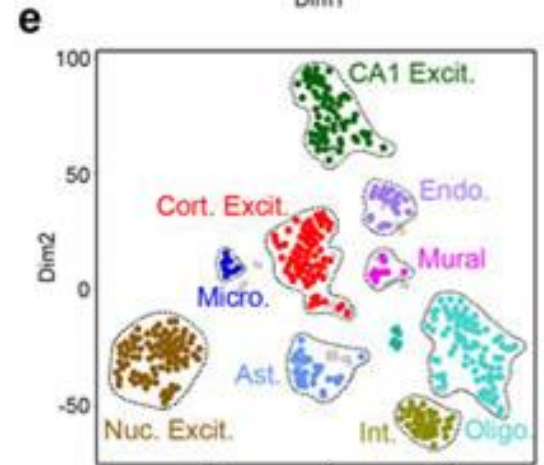
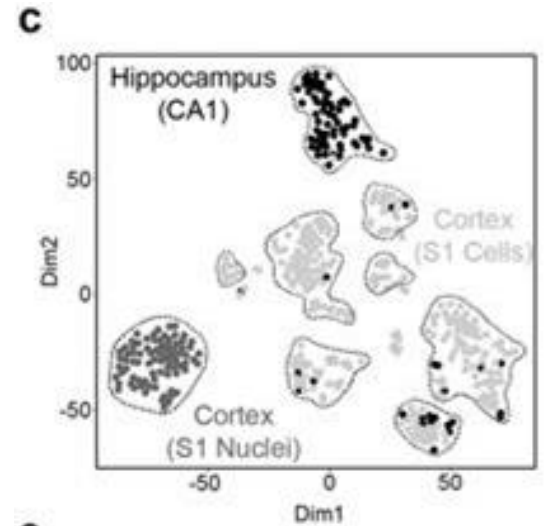
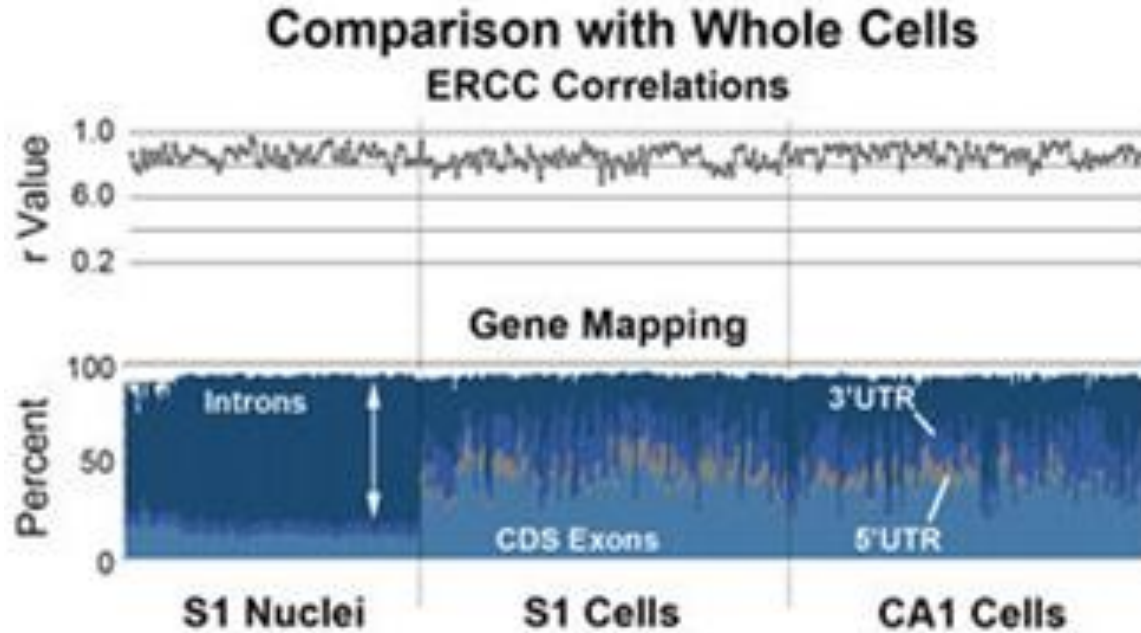
**What are the possible sources of intronic reads?**

**How would you quantify intronic reads?**

[GTF files usually contain “gene”, “UTR”, “transcript” and “exon” entries]



# Single-nuclei sequencing



# RNA Velocity

Assumes: Reads spanning into-exon boundary come from newly synthesized mRNAs which have not finished splicing

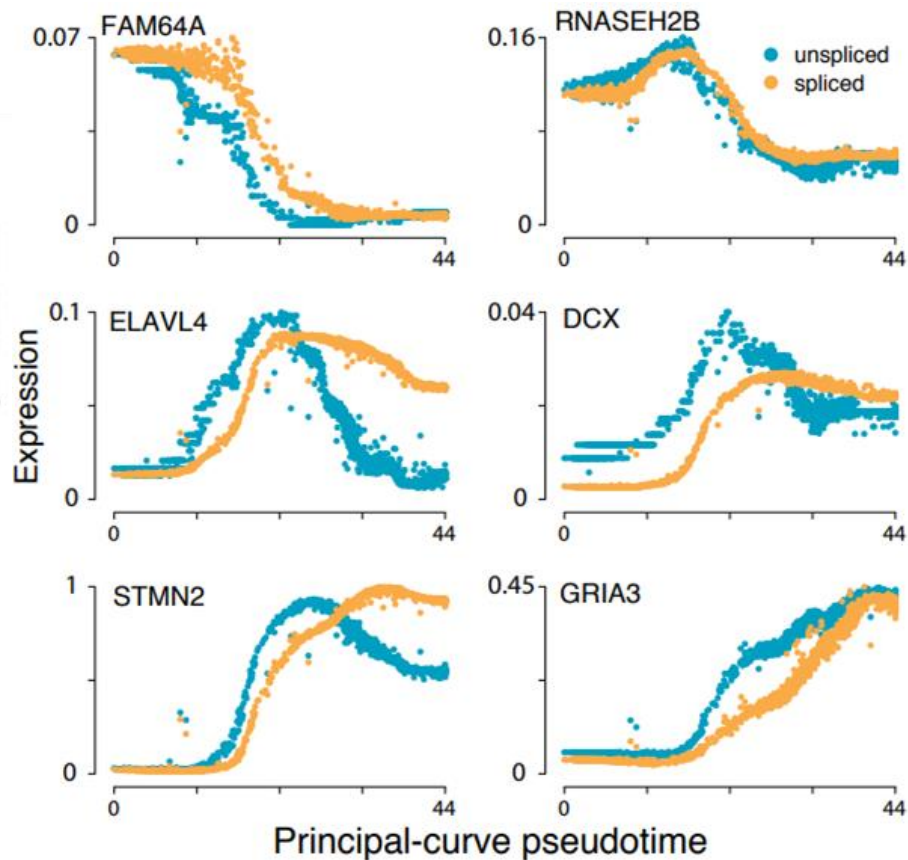
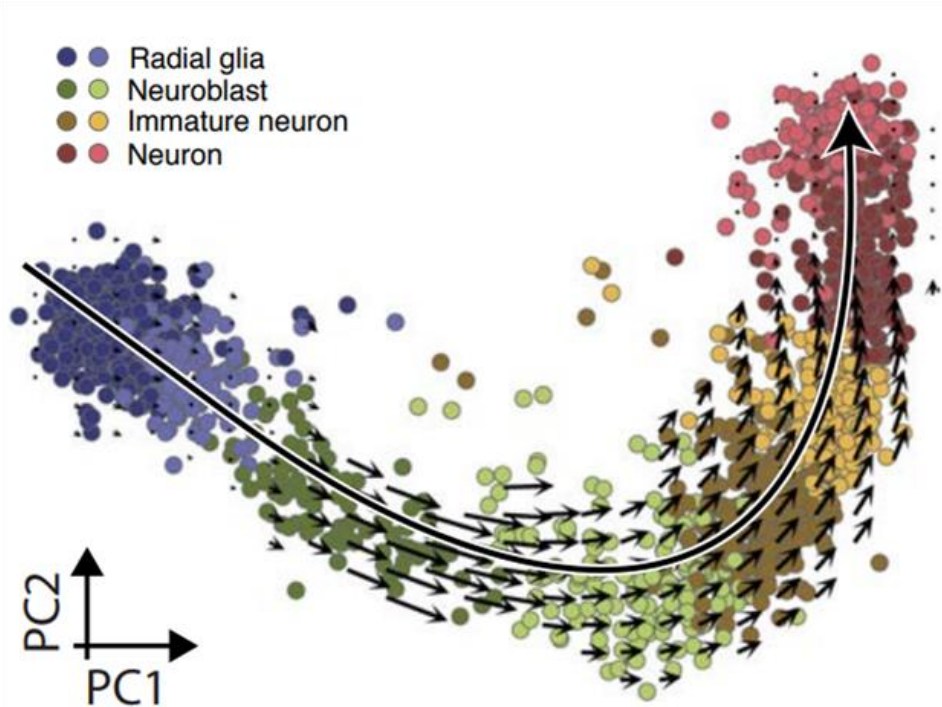
Completely intronic reads:

- Could be non-specific background (particularly repetitive elements)
- Could be unannotated exons/transcripts

At steady state:  $S_s = yU - d$

If  $S > S_s$  then expression level is decreasing and vice versa.

# RNA Velocity



Any Questions?

# Correcting for Gene length

CPM = counts per million =  
 $x / \text{colSums}(x) * 1000000$

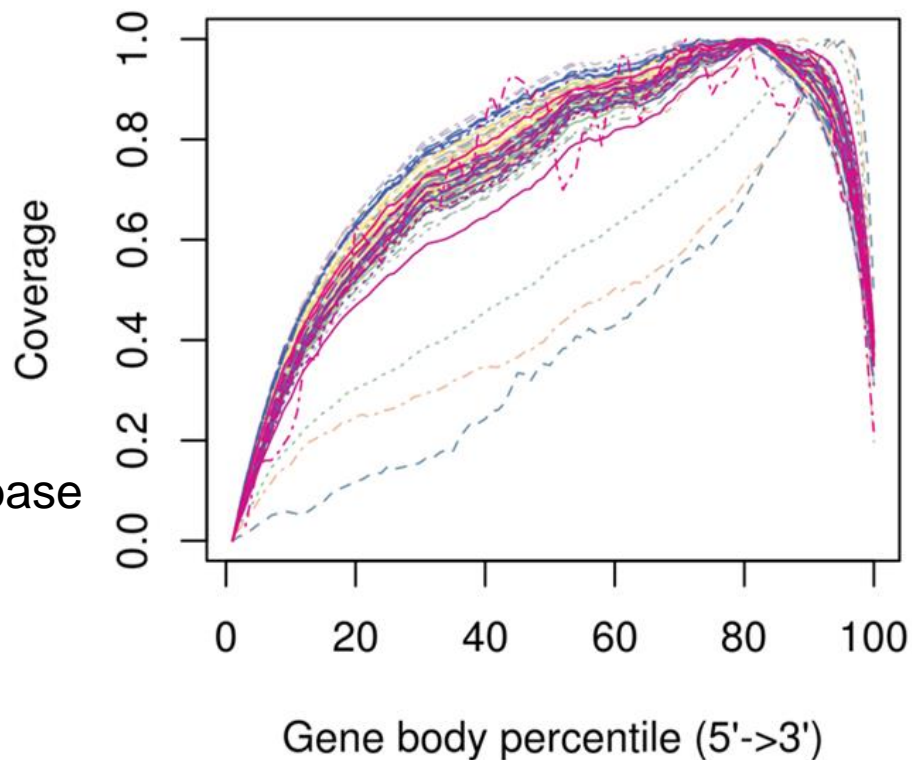
TPM = transcripts per million =  
 $\text{CPM}(x / \text{glength})$

CPM of UMI counts may also be referred to as TPM

FPKM/RPKM = fragments/reads per kilobase  
per million =  $\text{CPM} / \text{glength}$

Calculating “gene length” depends on:

- Isoform usage
- Gene body coverage



# Quantification methods and their units

Cufflinks : assembles transcripts, quantifies isoforms or genes, tests DE.

Due to uneven coverage may assemble “novel” 3’ transcripts.

This can be mitigated by providing a gtf to be tiled with faux-reads

Input: BAM file of reads aligned to the genome.

Units = FPKM/RPKM

RSEM : maps to transcriptome with STAR or bowtie2, “single-cell prior” option.

Quantifies isoforms or genes.

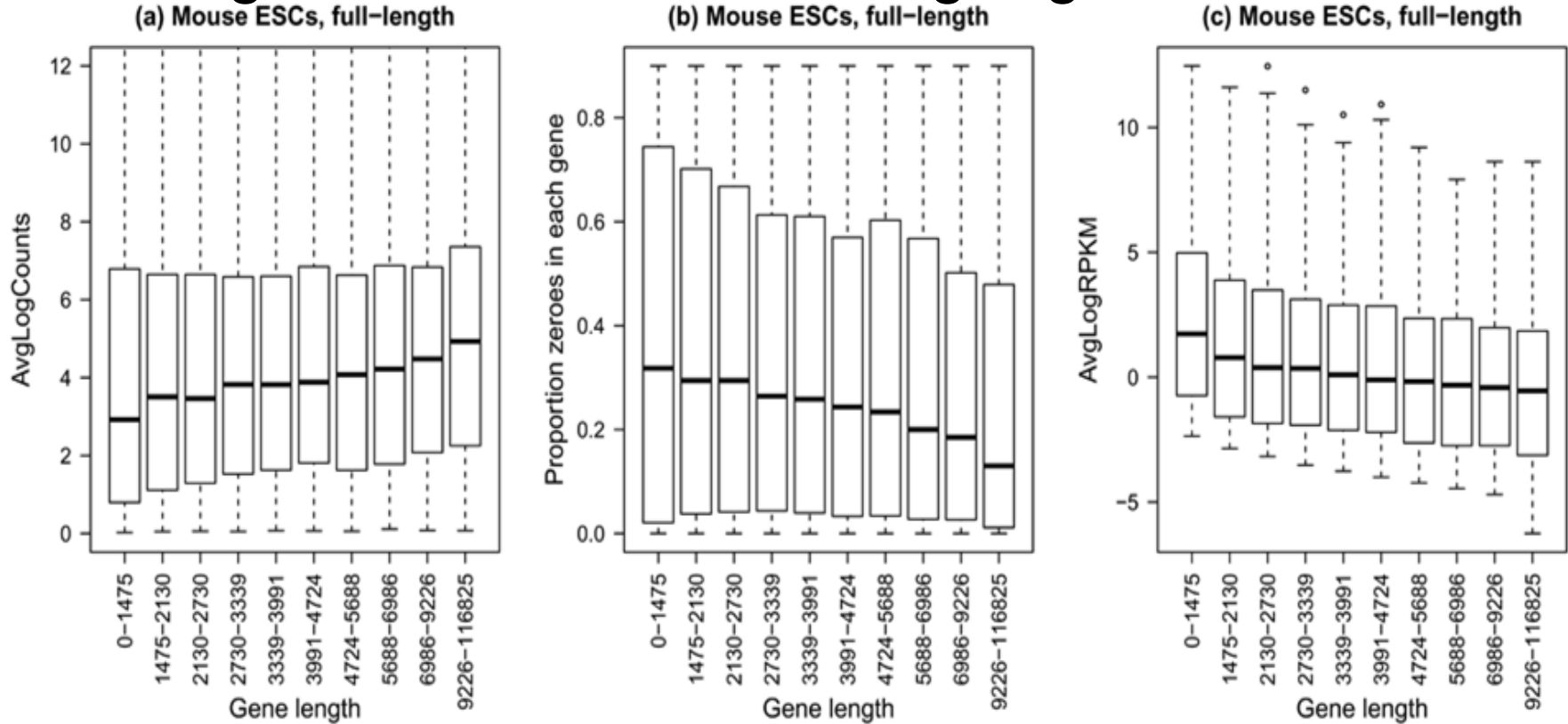
Units = TPM or FPKM

Kallisto/Salmon : pseudo-alignment estimates expression of transcripts directly

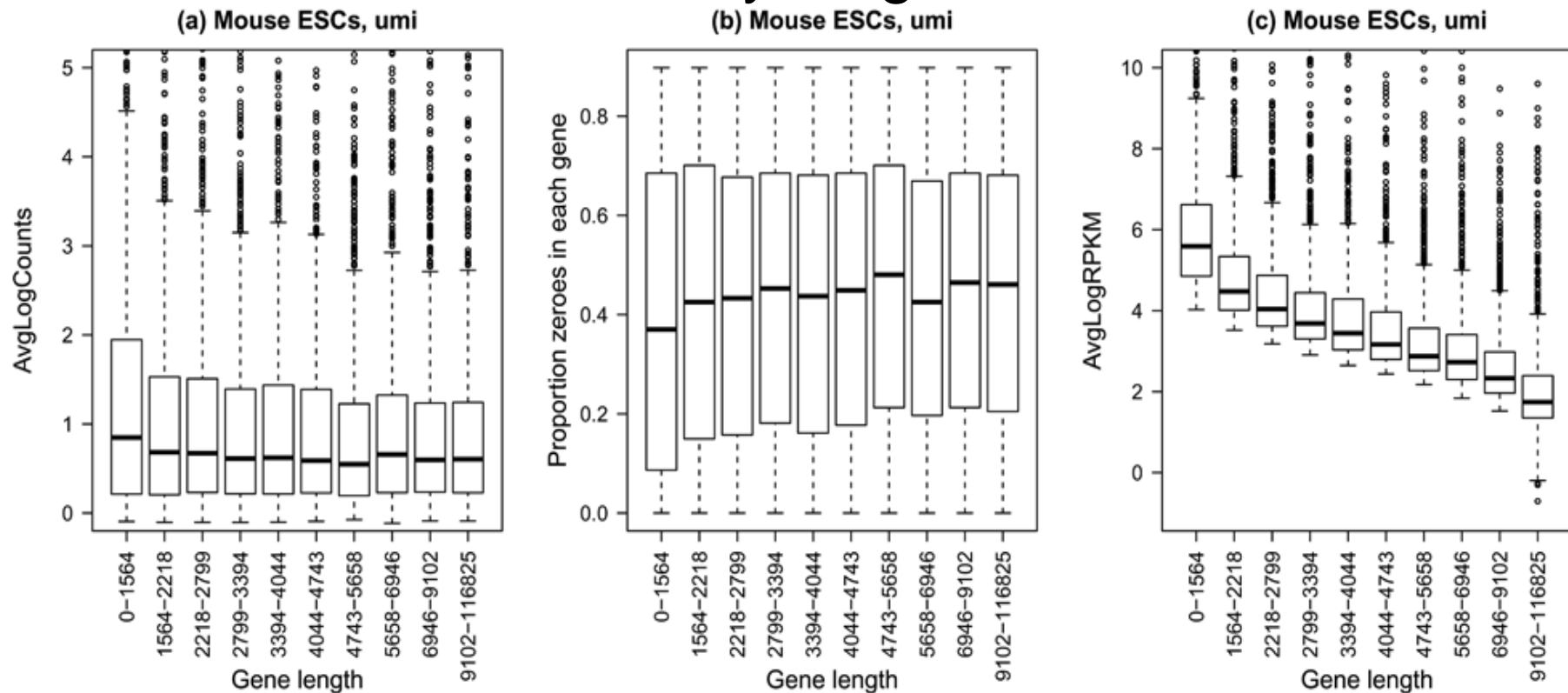
Units = Counts or TPM



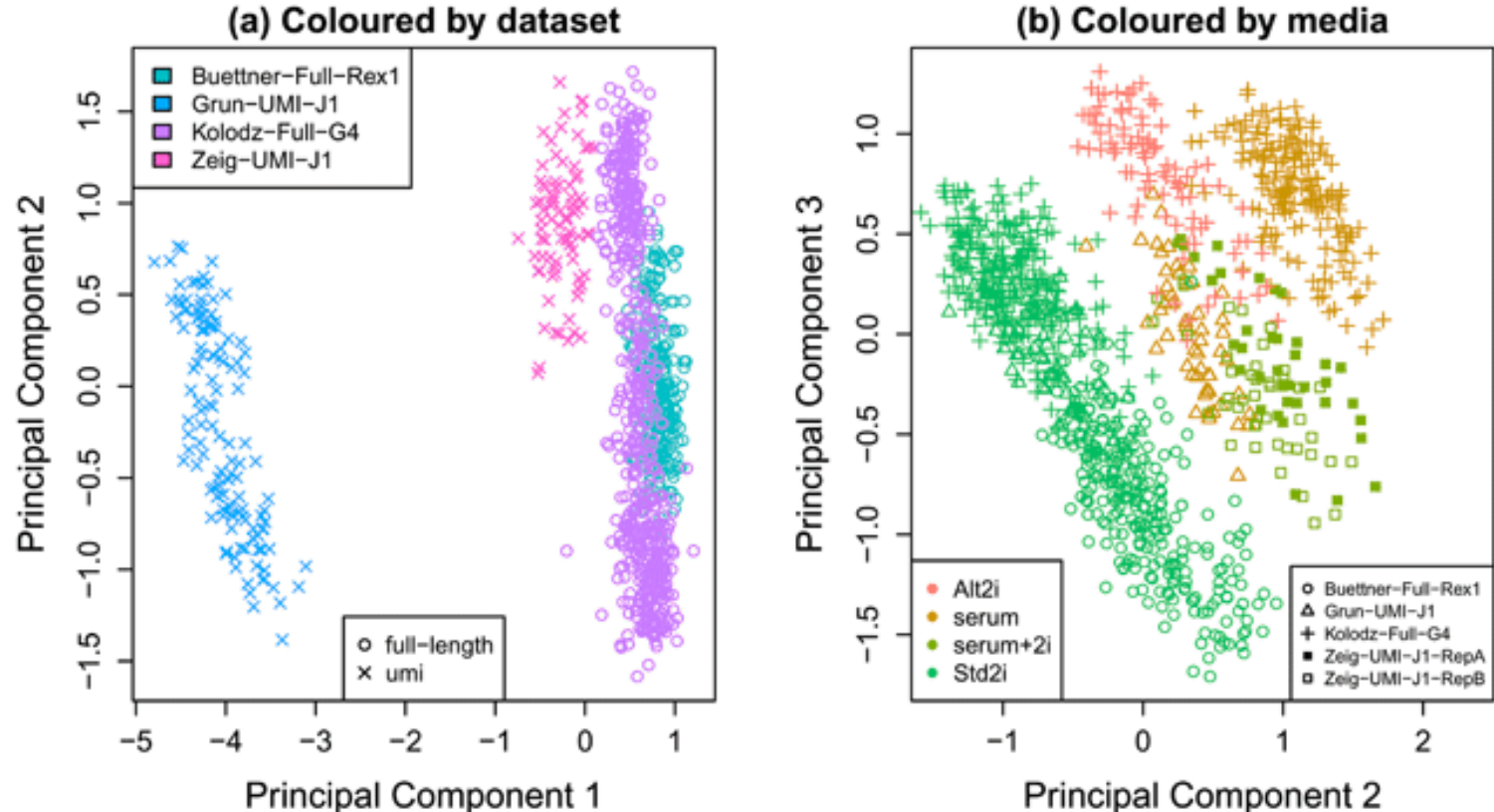
# Full length data is biased to longer genes



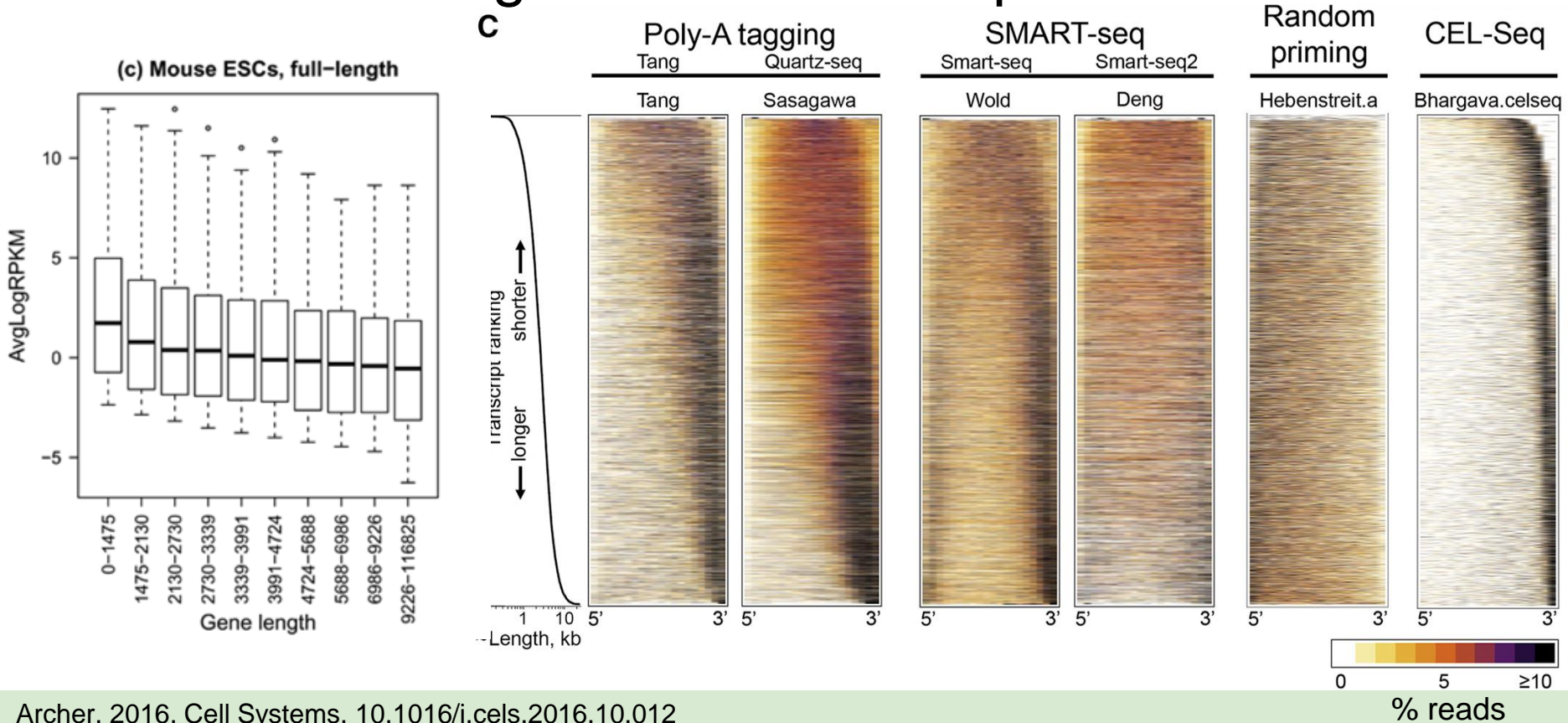
# UMI data is not biased by length



# UMI and Full-transcript data are not the same



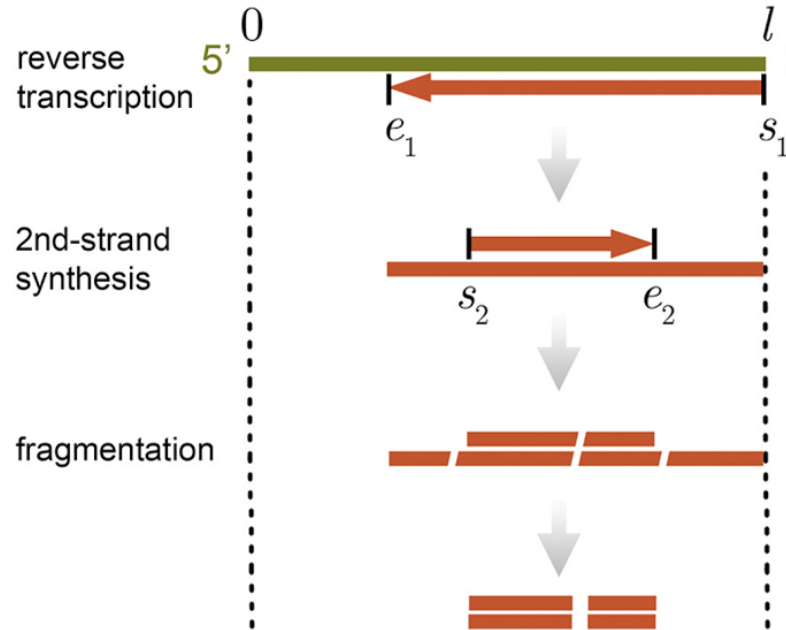
# Uneven coverage violates assumptions of RPKM





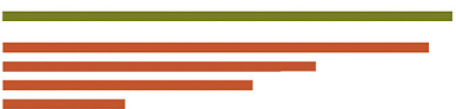


# Uneven coverage due to incomplete RT

**A**

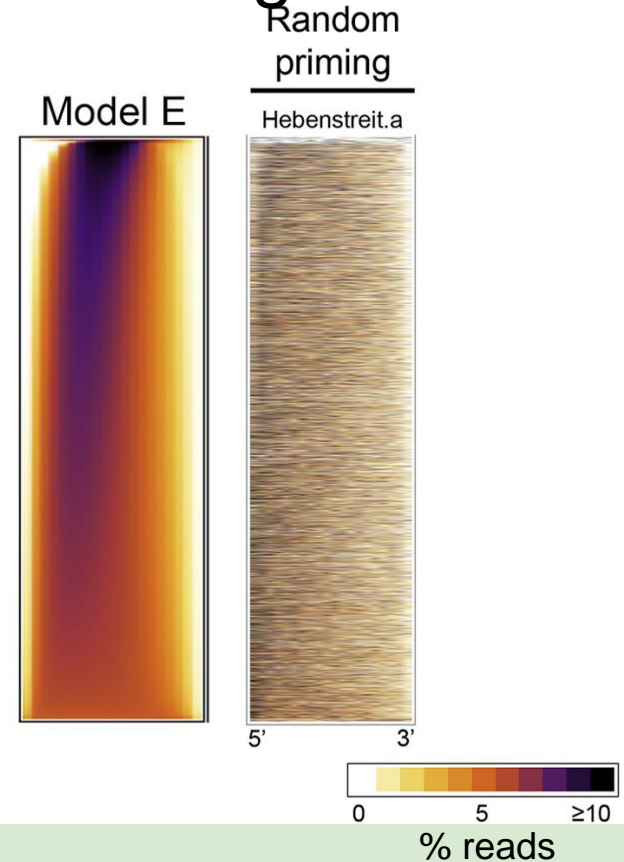
Typical scRNA-seq protocol  
(non-RNA-fragmentation)



# Uneven coverage due to incomplete RT

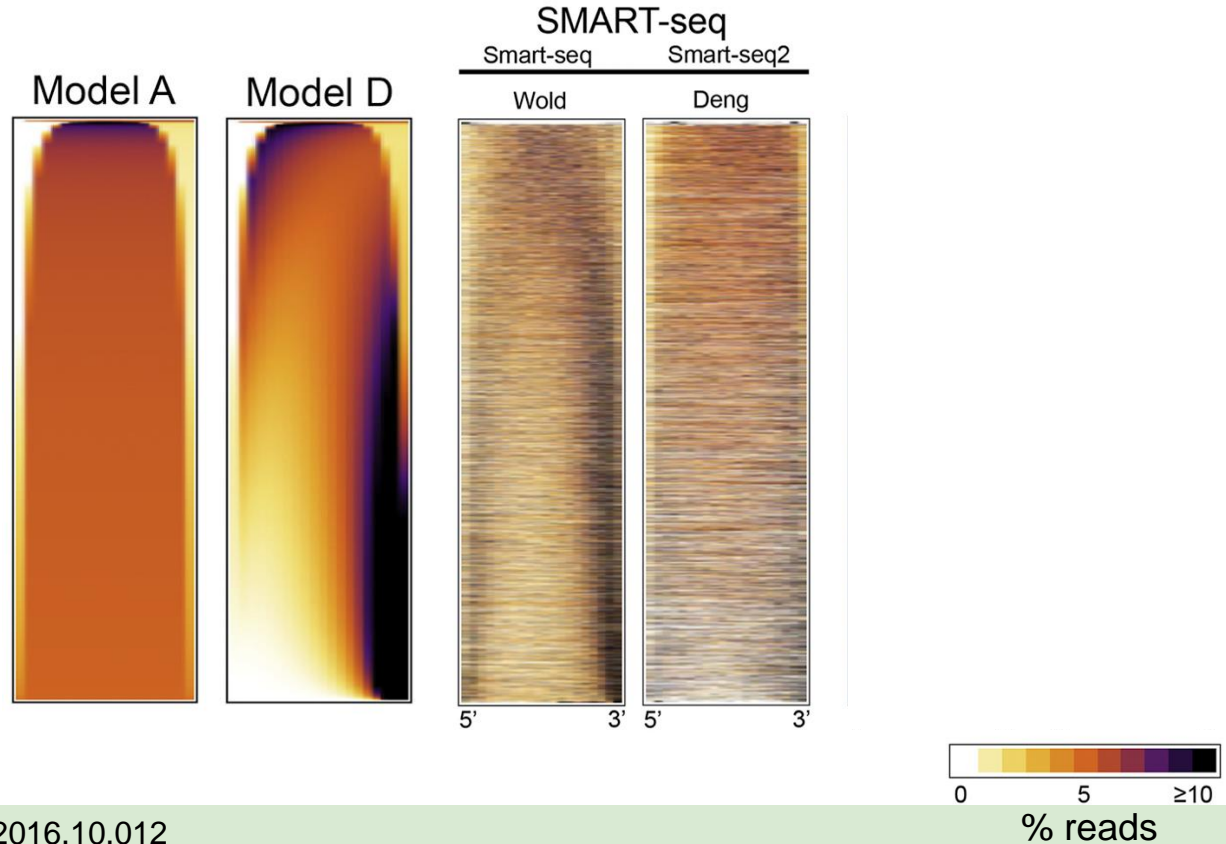
Model/Scenario	Priming and synthesis	Compatible with assumption of	Example scheme
A	terminally primed <i>complete</i> 1st-strand terminally primed <i>complete</i> 2nd-strand	ideal SMART	
B	terminally primed <i>incomplete</i> 1st-strand terminally primed <i>complete</i> 2nd-strand	ideal poly-A tagging non-ideal SMART	
C	terminally primed <i>complete</i> 1st-strand terminally primed <i>incomplete</i> 2nd-strand	non-ideal SMART non-ideal poly-A tagging	
D	terminally primed <i>incomplete</i> 1st-strand terminally primed <i>incomplete</i> 2nd-strand	non-ideal SMART non-ideal poly-A tagging	
E	internally primed <i>incomplete</i> 1st-strand internally primed <i>incomplete</i> 2nd-strand	random priming	
mixture model	combination of above	non-ideal various	?

# Model recapitulates observed gene coverage



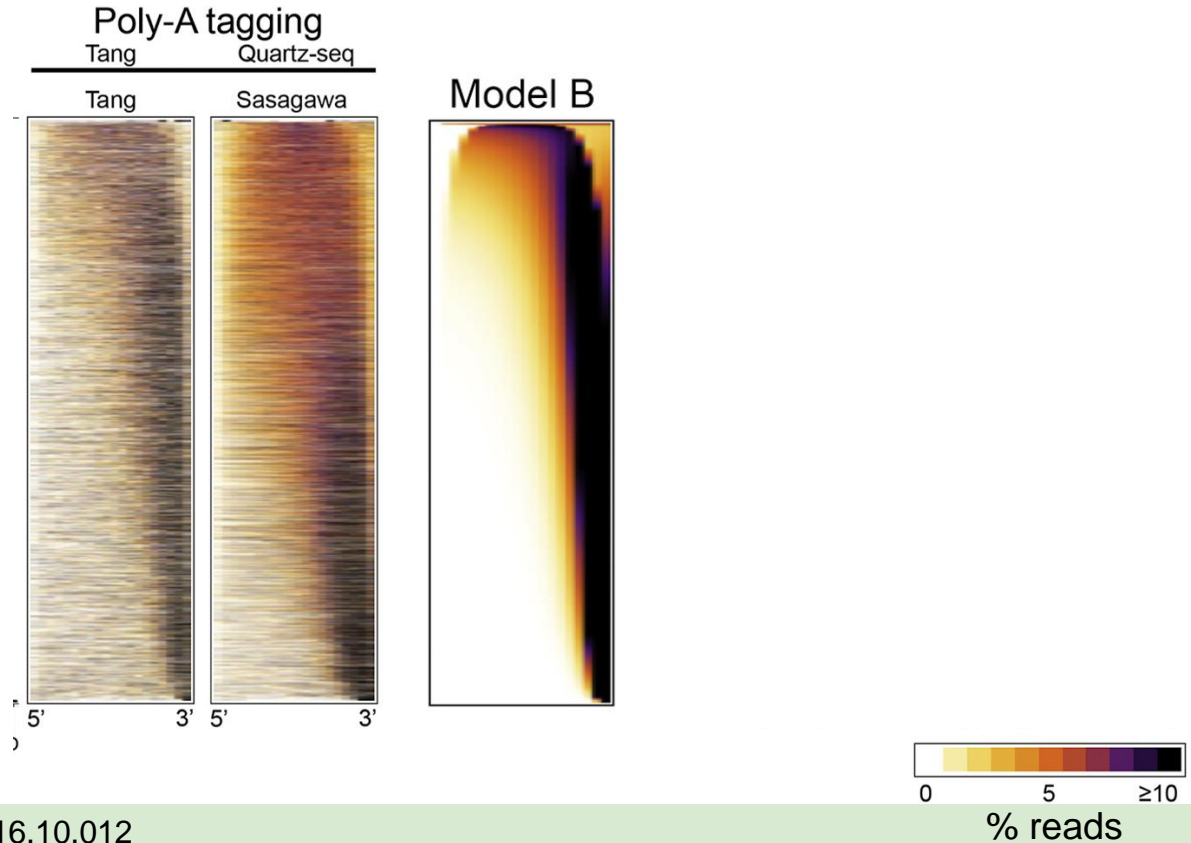


# Model recapitulates observed gene coverage





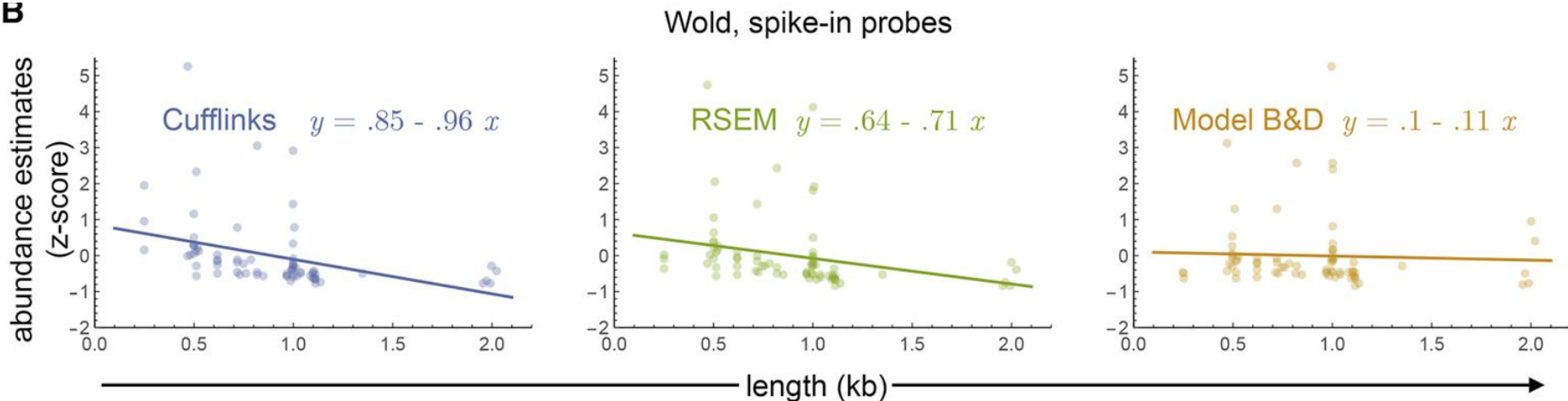
# Model recapitulates observed gene coverage



# Model-based gene-length correction (Libinorm)

Libinorm outperforms Cufflinks and RSEM for Smartseq scRNASeq data.

**B**



Questions?

# Isoform Quantification

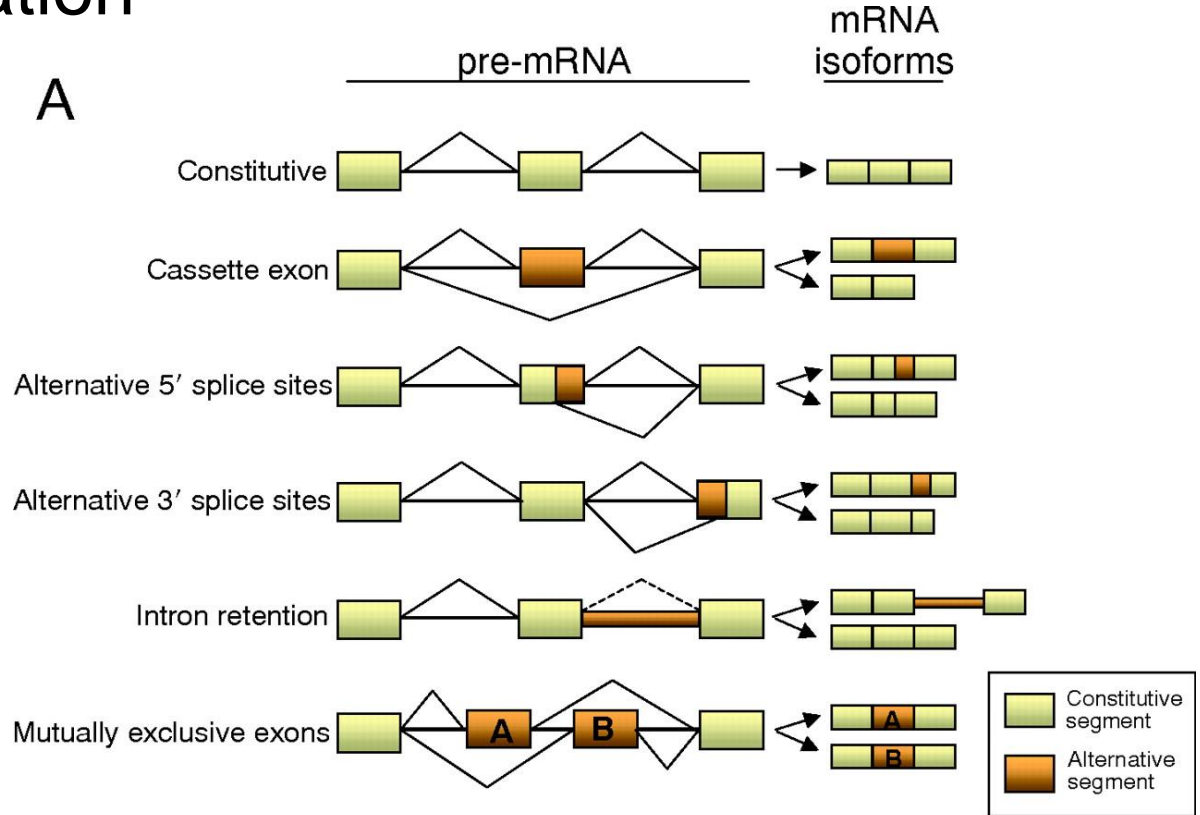
No single-cell specific tools.

A

Bulk RNASeq tools:

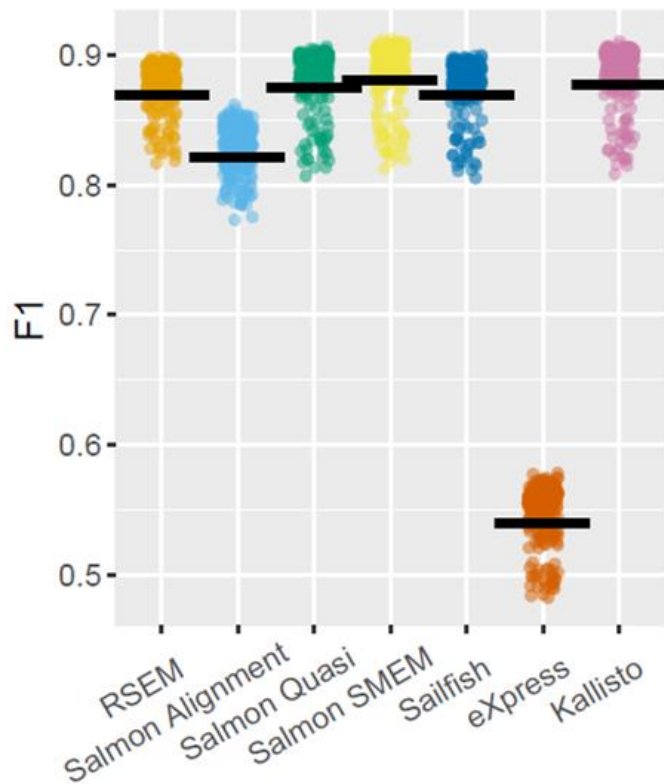
- RSEM
- Cufflinks
- Sailfish
- eXpress
- Kallisto
- Salmon

Use both split reads and relative expression level

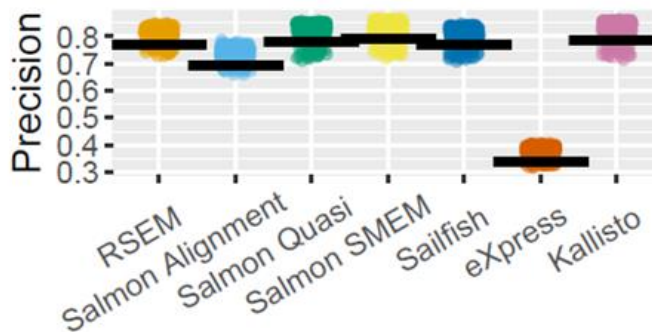


# Isoform Quantification - Full Transcript (7m reads/cell)

A F1



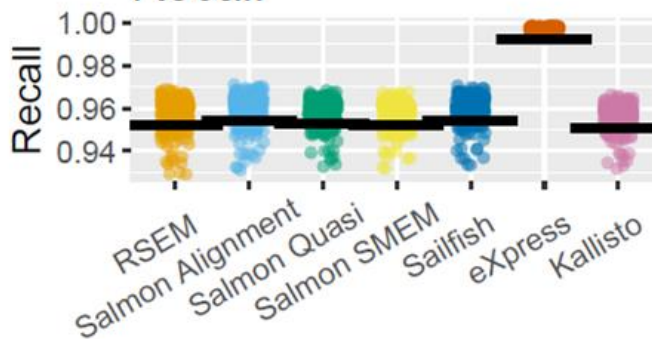
Precision



Presence /  
Absence

Precision =  
 $TP / (FP + TP)$

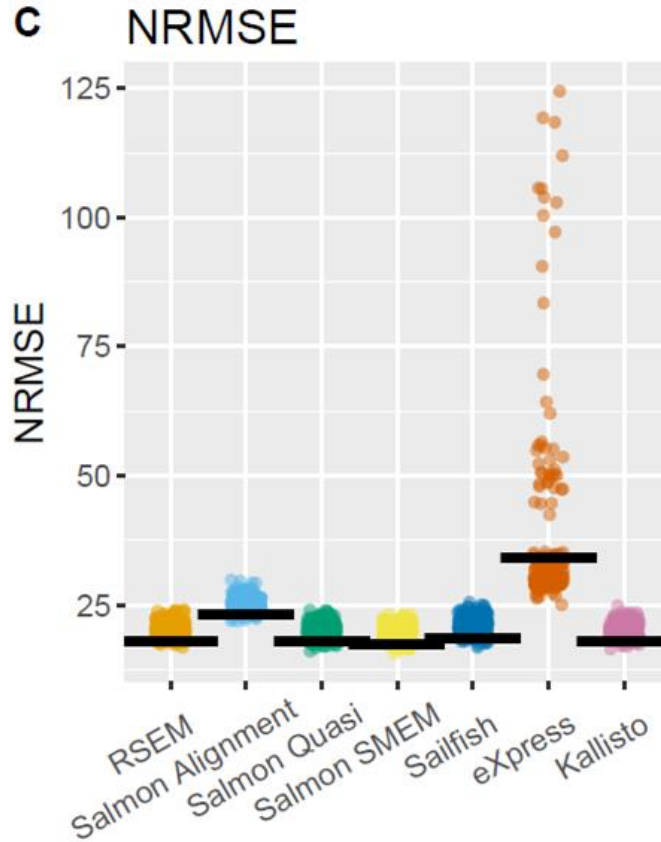
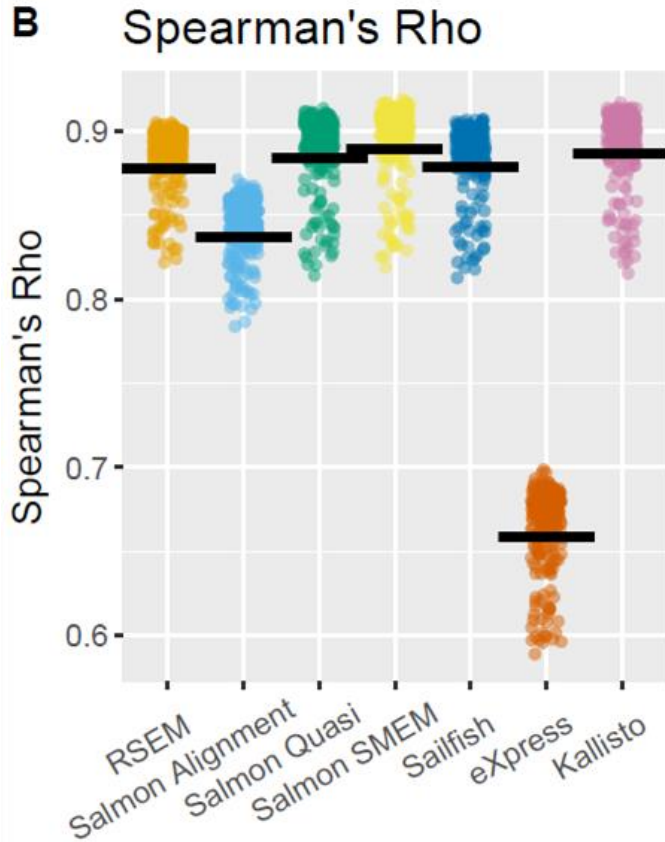
Recall



Recall =  
 $TP / (TP + FN)$

F1 =  $2 / (Precision^{-1} + Recall^{-1})$

# Isoform Quantification - Full Transcript (7m reads/cell)

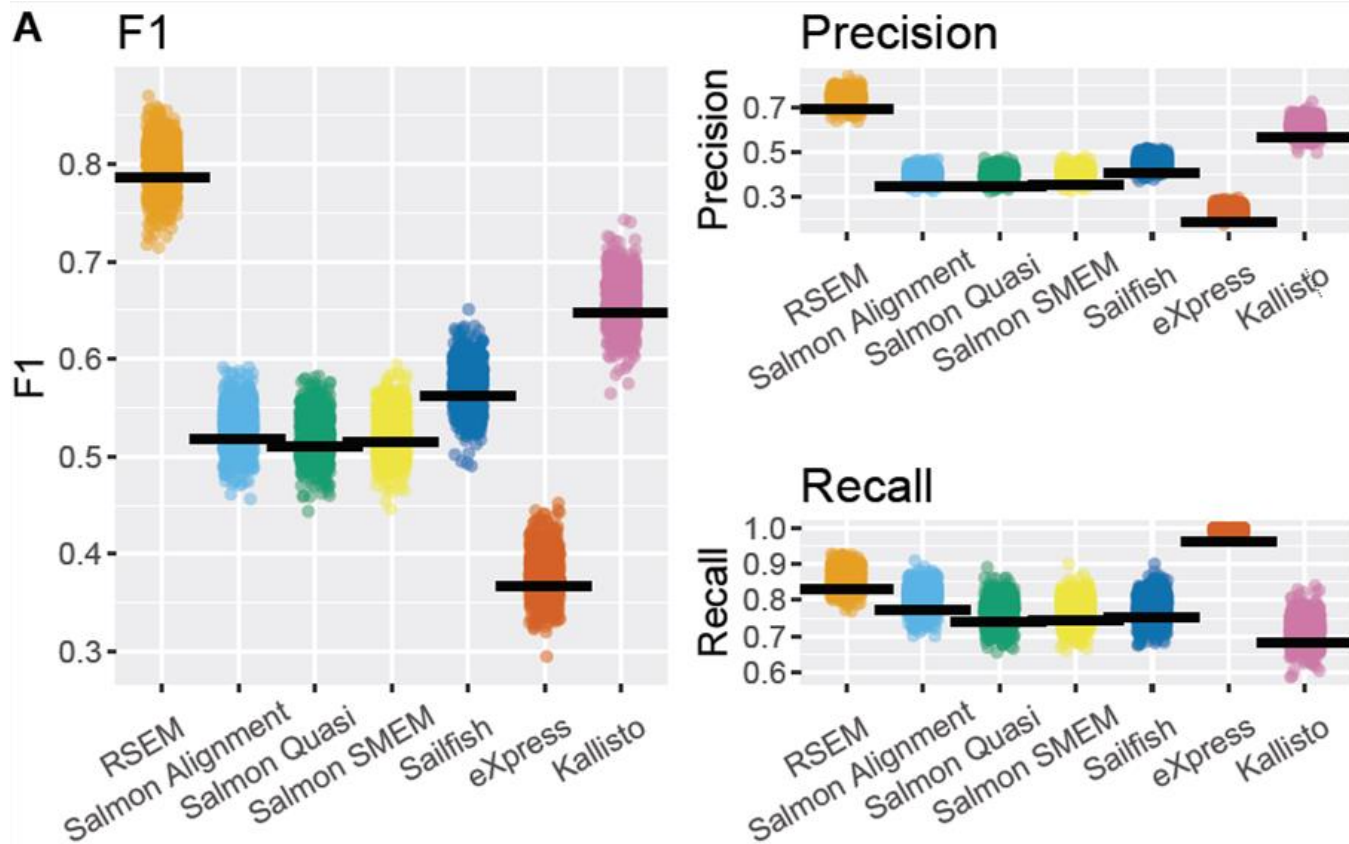


Accurate  
Quantification

Spearman's Rho  
= rank correlation

NRMSE = error  
from  $y=x$

# Isoform Quantification - Drop-seq (8k reads/cell)



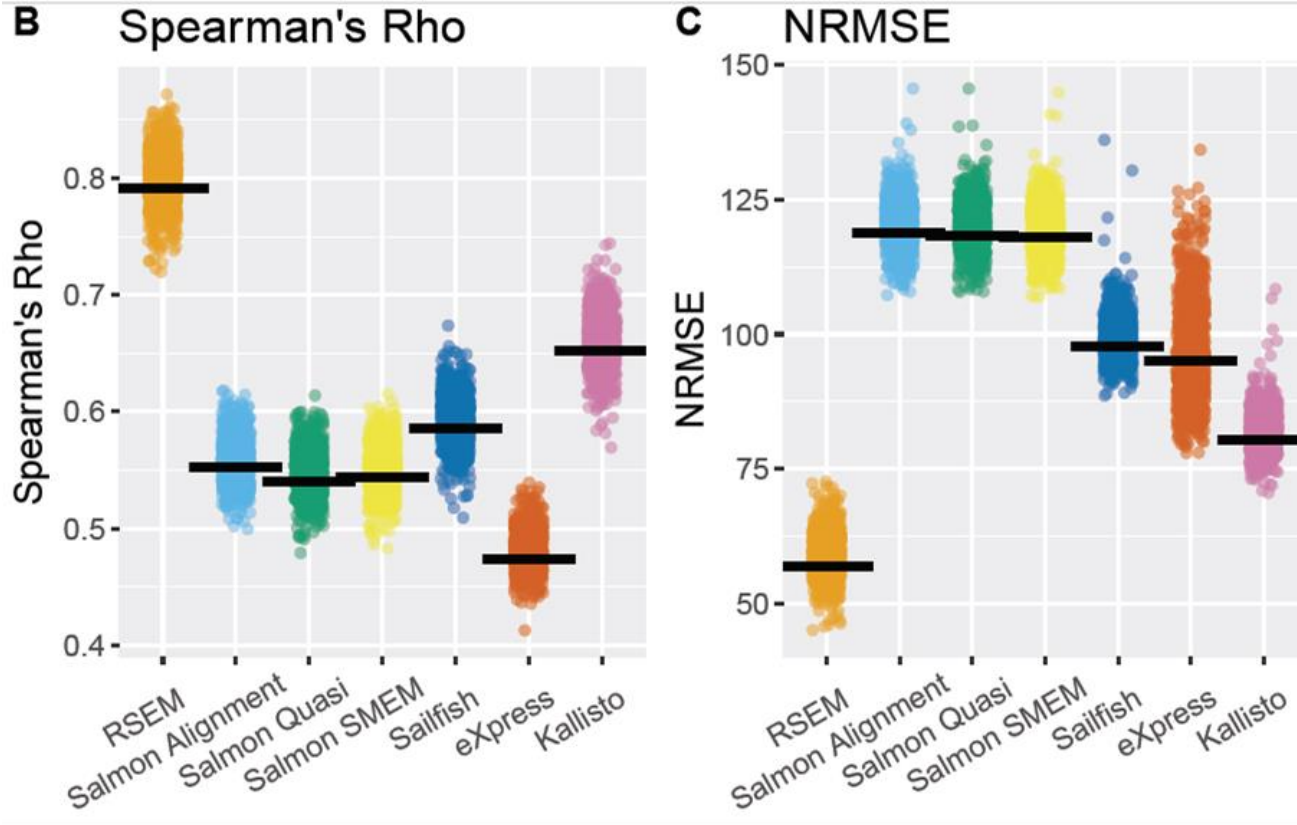
Presence /  
Absence

Precision =  
 $TP / (FP + TP)$

Recall =  
 $TP / (TP + FN)$

F1 =  $2 / (Precision^{-1} + Recall^{-1})$

# Isoform Quantification - Drop-seq (8k reads/cell)



Accurate  
Quantification

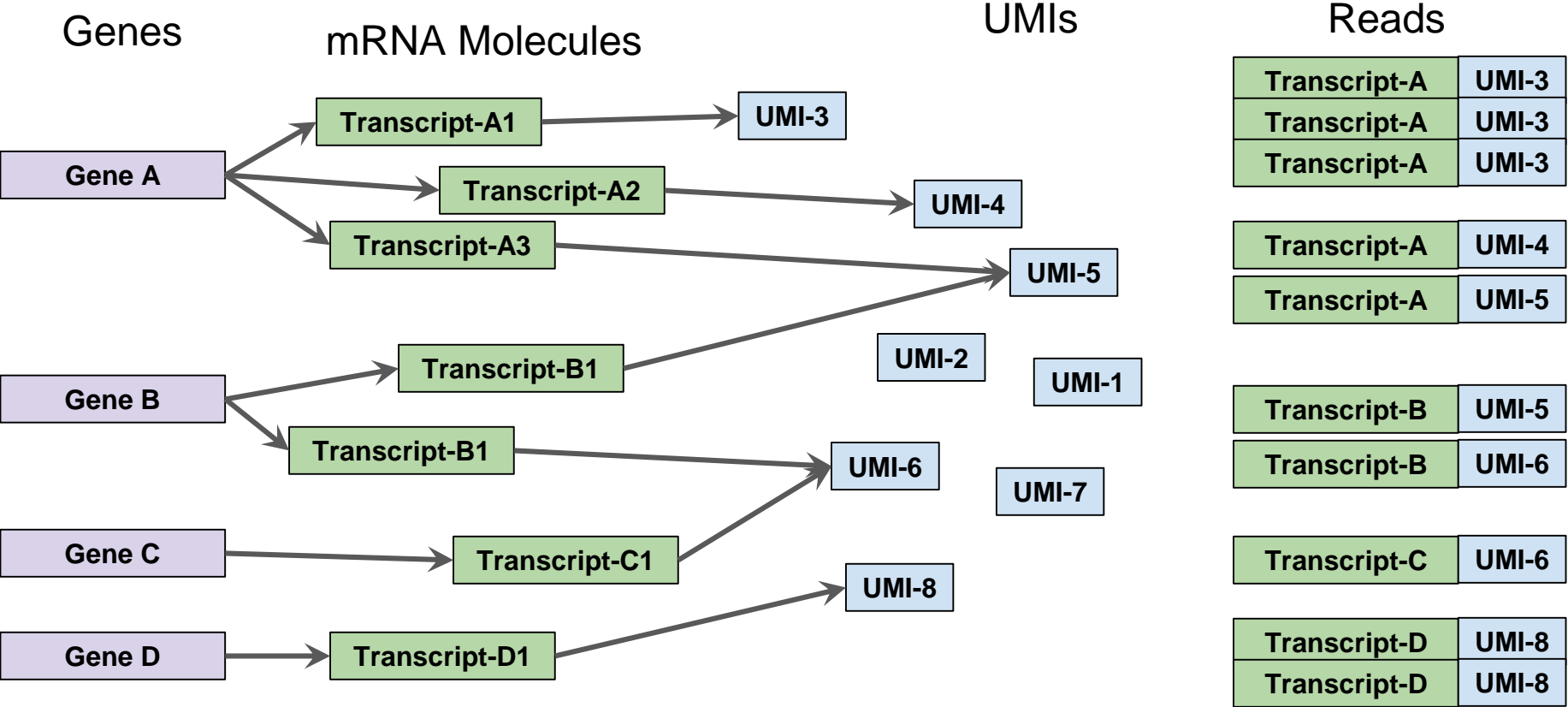
Spearman's Rho  
= rank correlation

NRMSE = error  
from  $y=x$

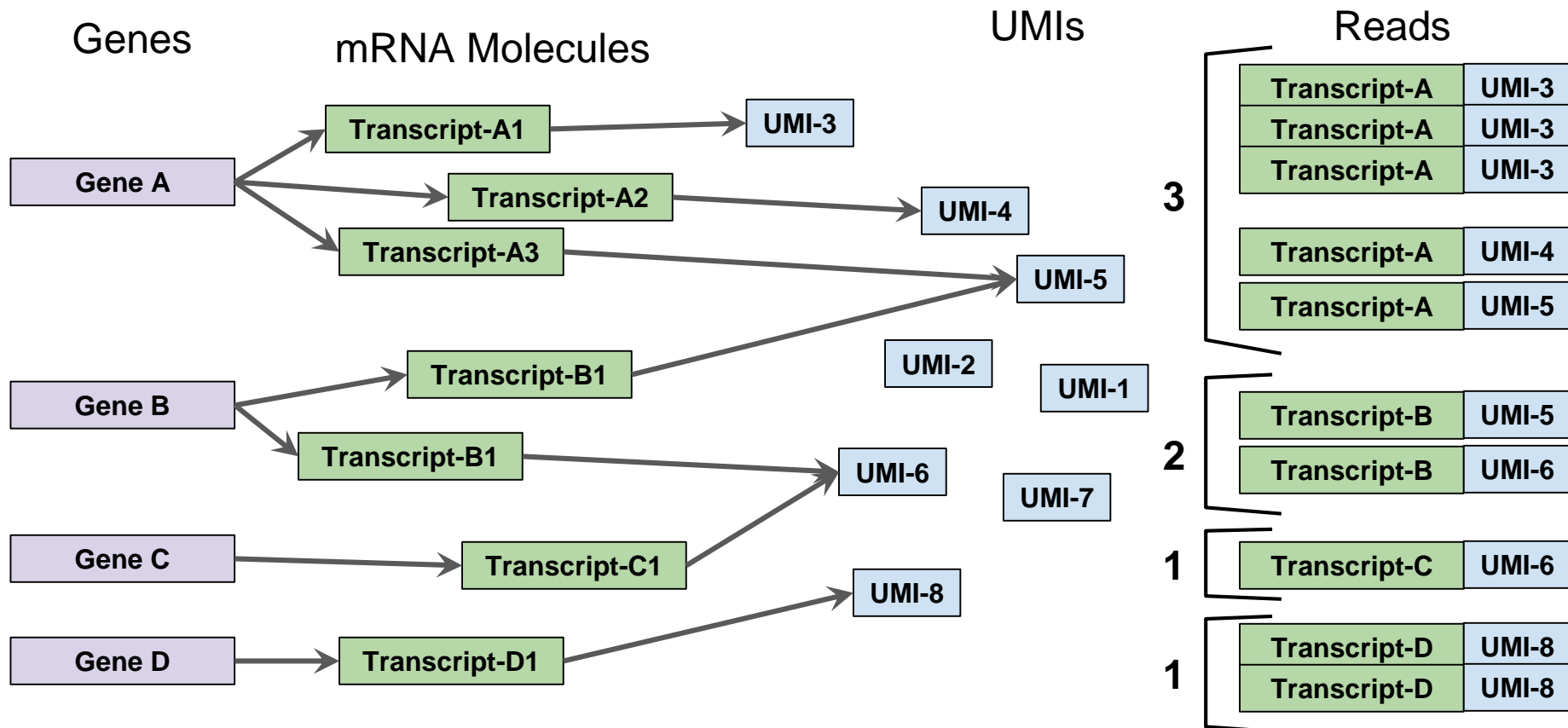


Any Questions?

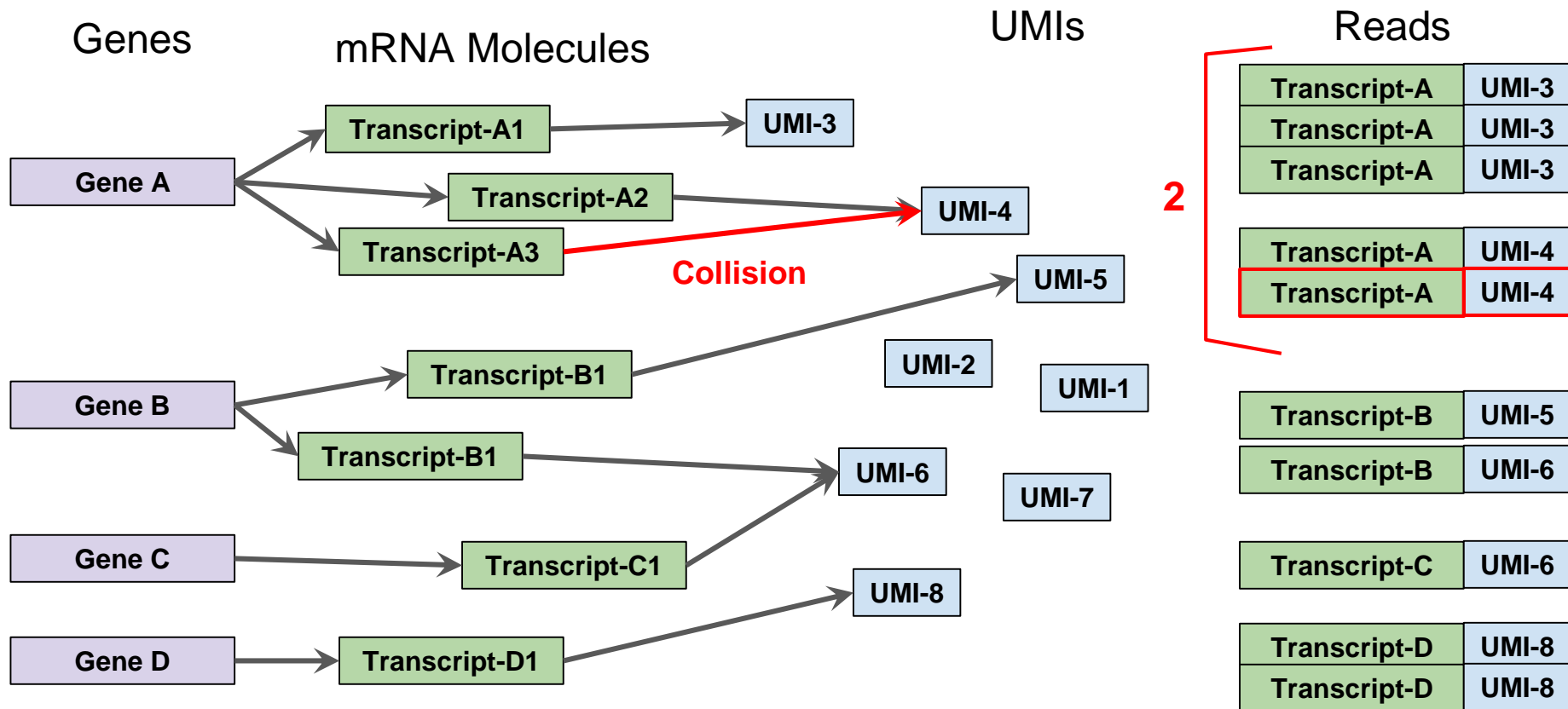
# Counting UMIs



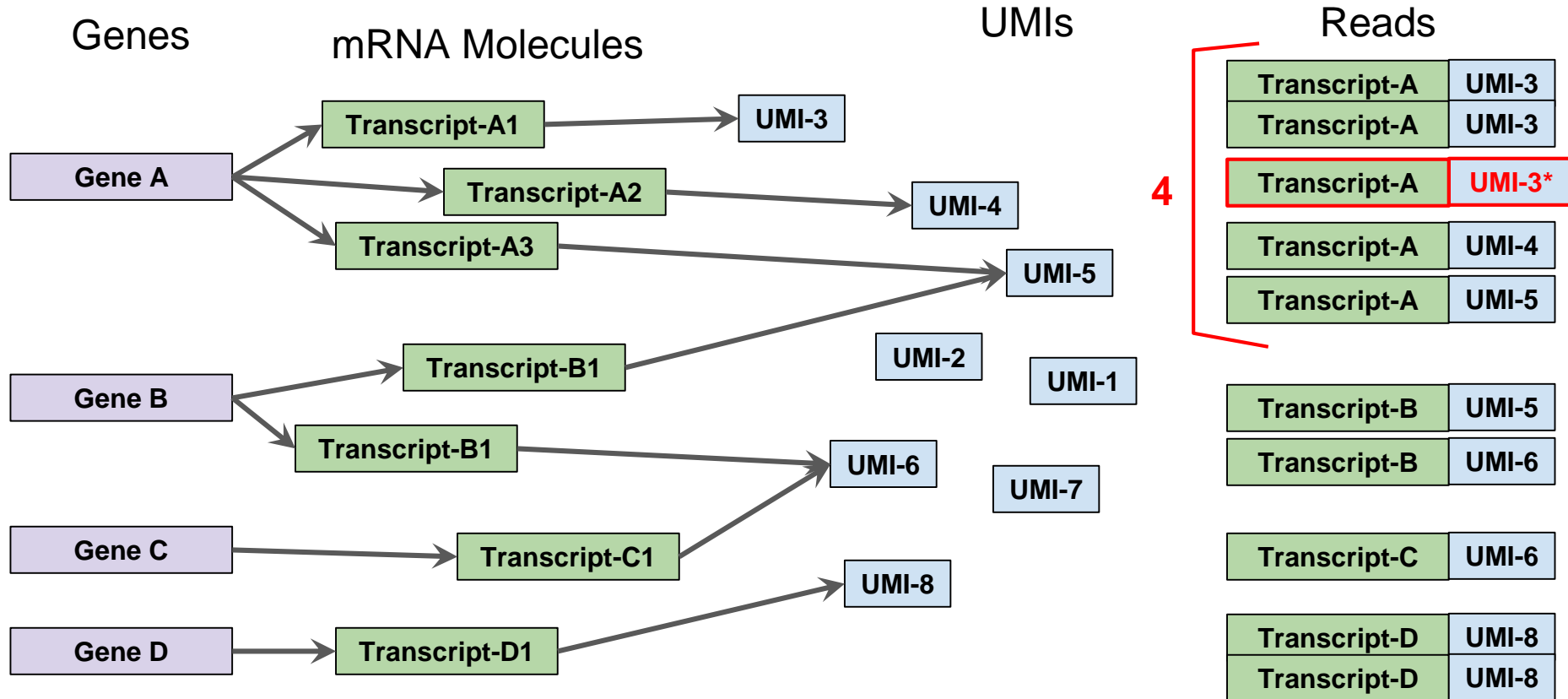
# Counting UMIs: What could go wrong?



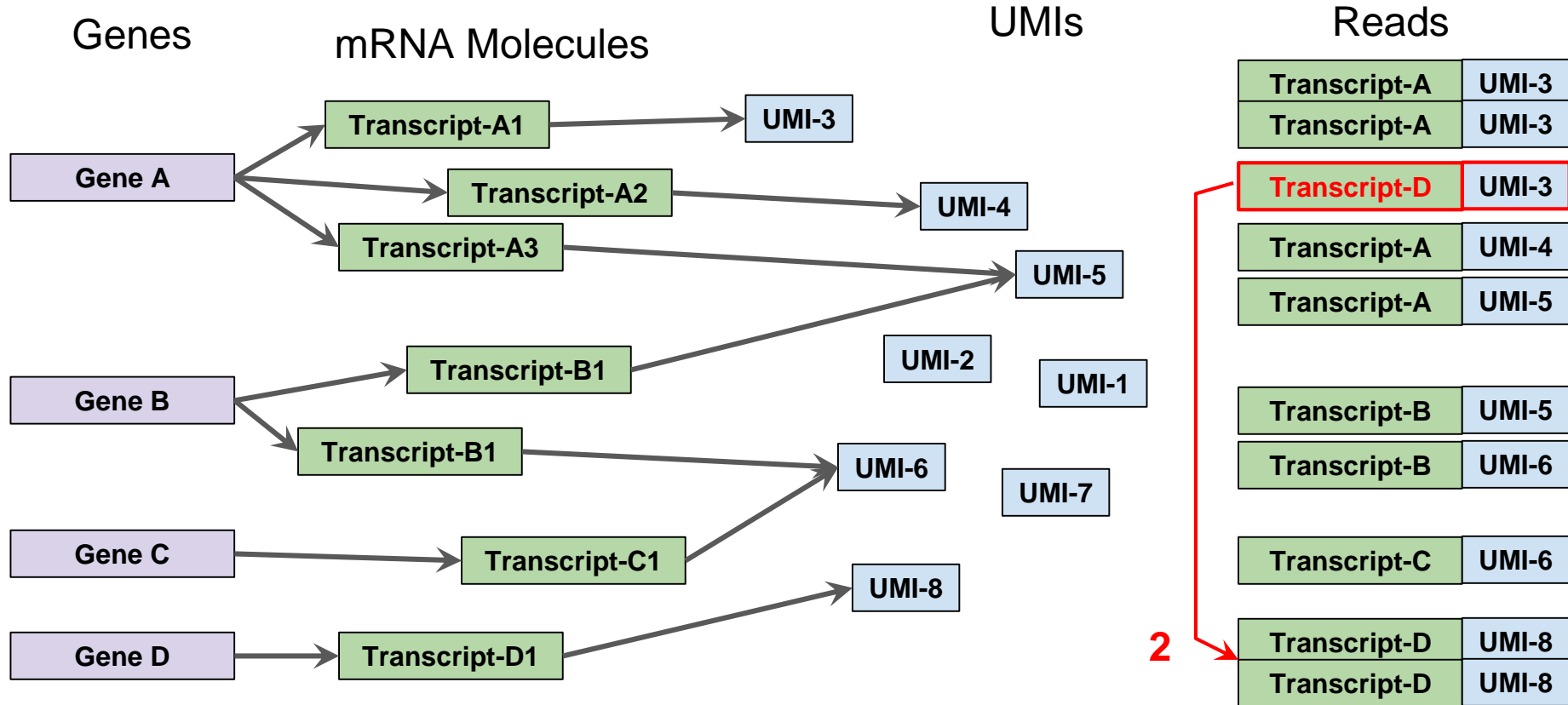
# Counting UMIs - Collisions



# Counting UMIs - Sequencing Errors



# Counting UMIs – Mismapping/Multimapping



# Discussion: How would you correct for these errors?

- Collisions
- Sequencing errors
- Mapping errors

# Correcting Collisions: Uniform UMIs

Assume  $n$  UMIs are equally frequent in the barcode pool, are used to tag  $m$  molecules of mRNA from a particular gene.

The number of mRNAs with any particular UMI is:  $C \sim \text{Poisson}(m/n)$

$$P(C > 0) = 1 - P(C=0) = 1 - \exp(-m/n)$$

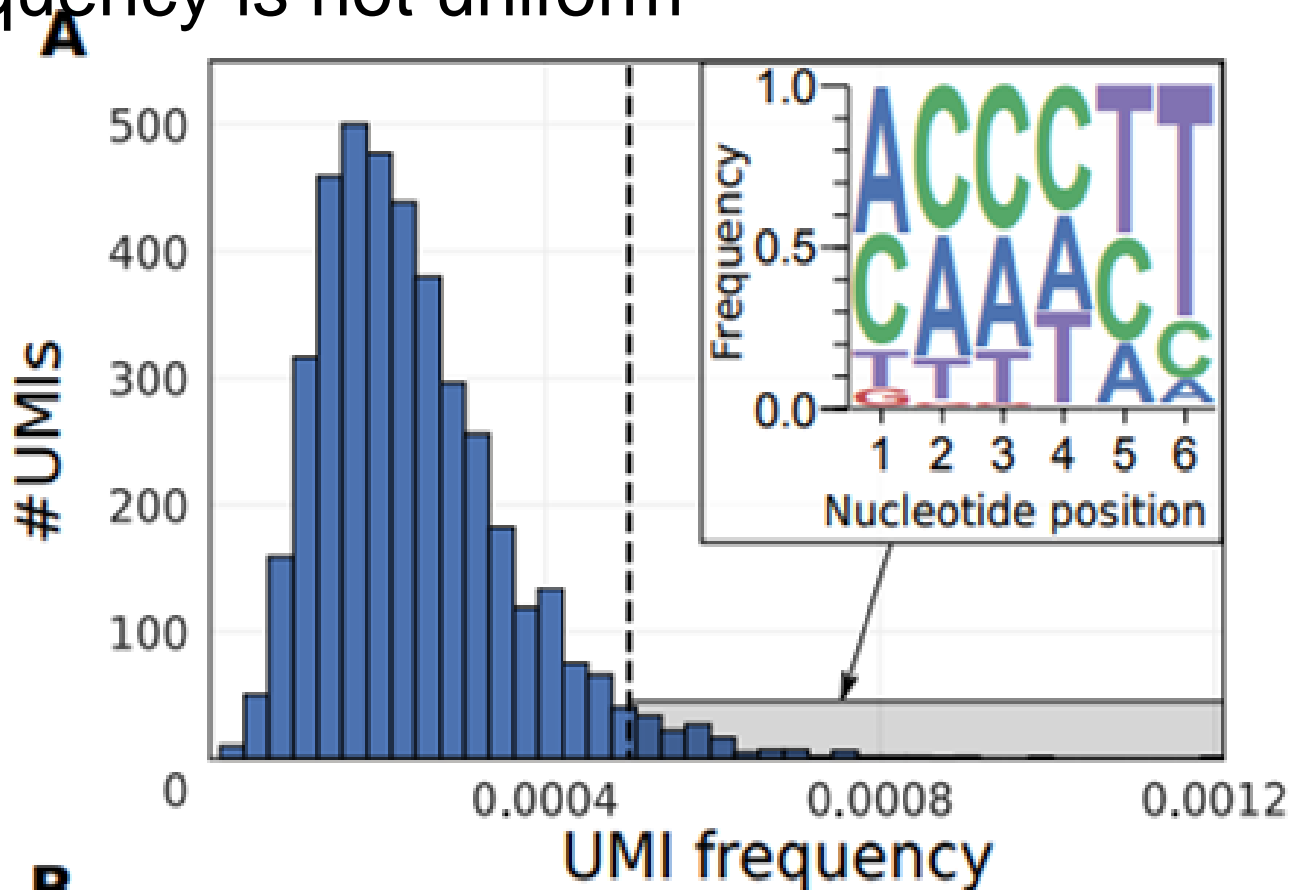
Expected number of observed barcodes is:  $b = n[1 - \exp(-m/n)]$

Rearranged to solve for  $m$ :

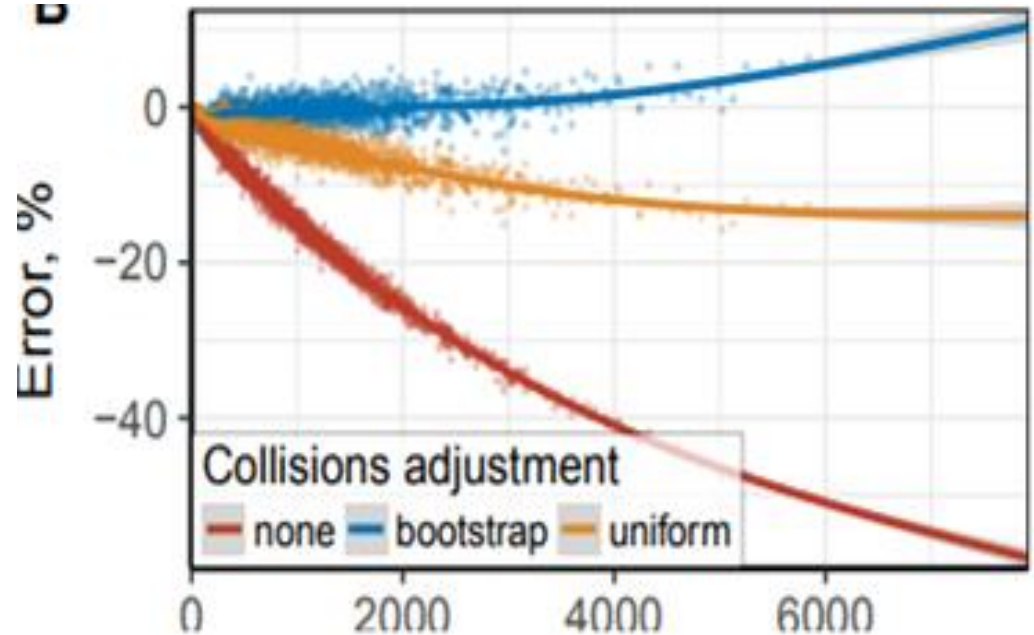
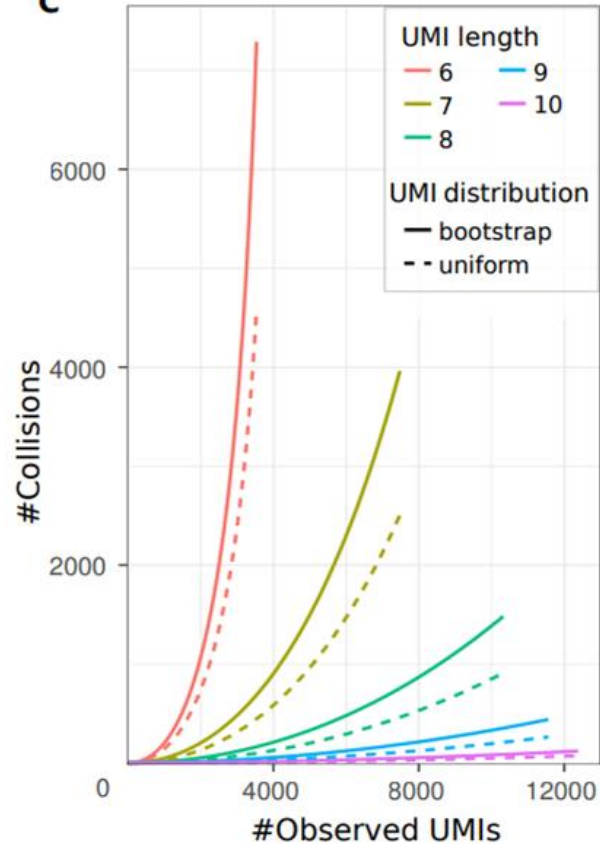
$$m = -n \log(1 - b/n)$$



# UMI frequency is not uniform



# Uneven UMI distribution correction (dropEst)



# Correcting For Sequencing Errors (UMI-Tools)

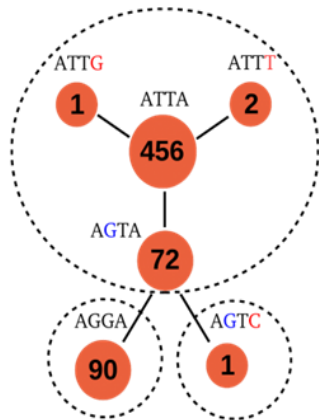
Two UMIs (ATTA & AGGA) with **PCR errors** and **sequencing errors**

Directional adjacency



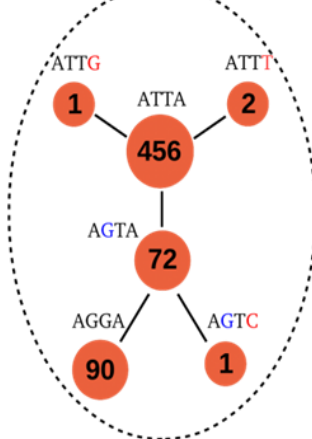
2

Adjacency



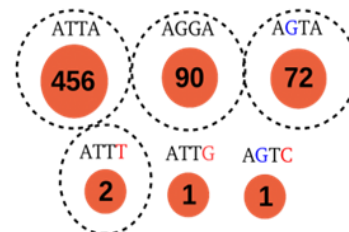
3

Cluster



1

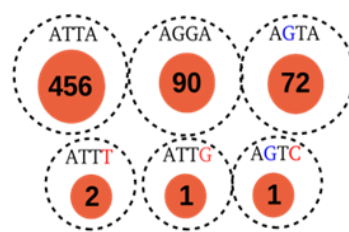
Percentile



Mean counts = 104  
1% threshold = 1.04

4

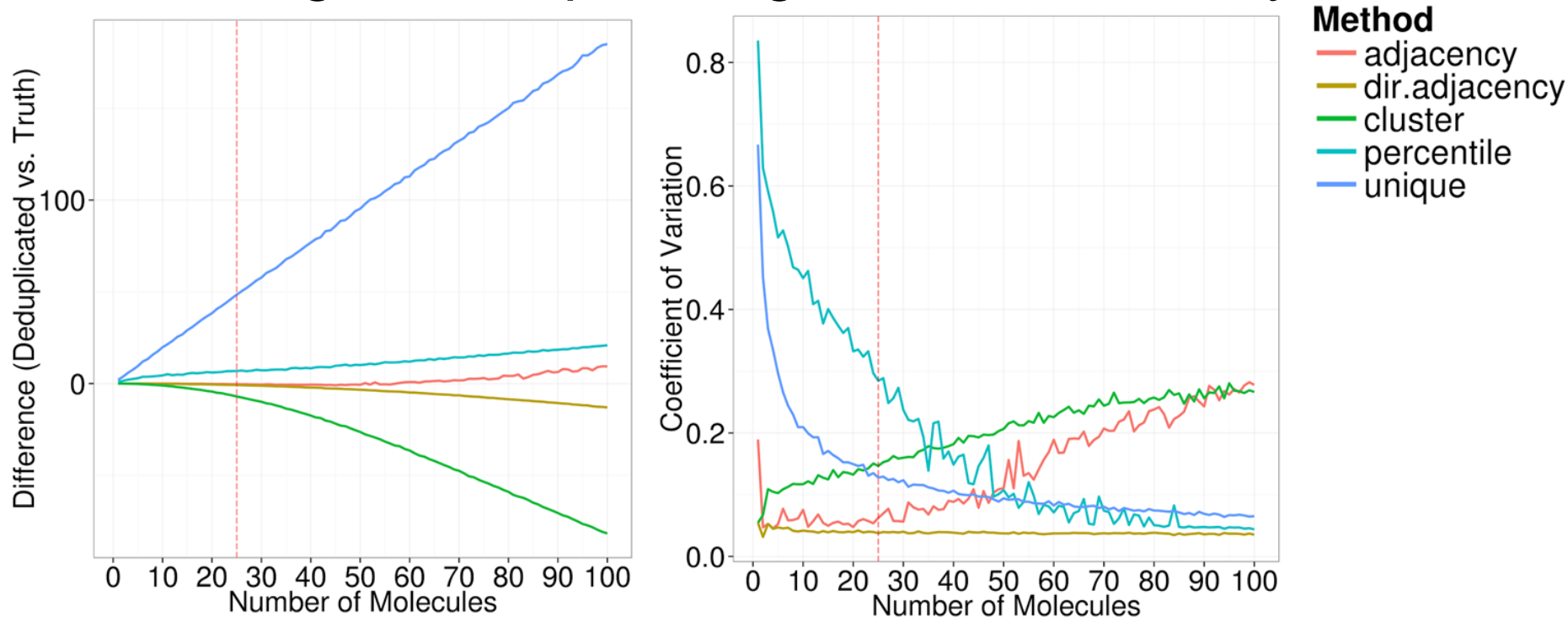
Unique



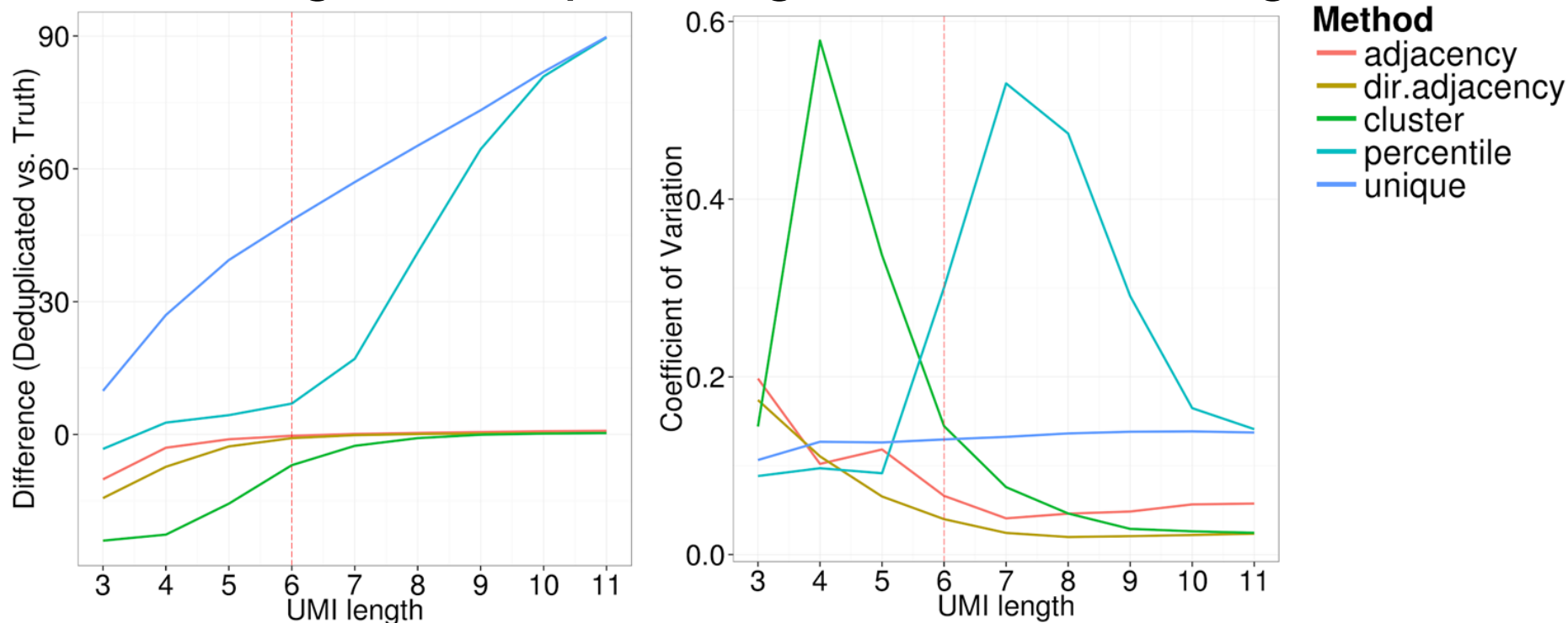
6

Number of inferred unique molecules

# Correcting For Sequencing Errors : Accuracy



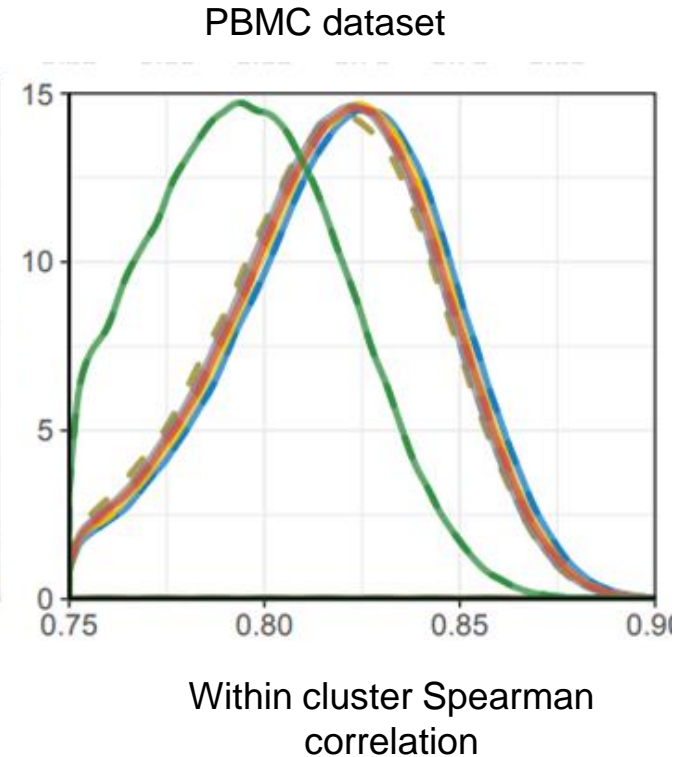
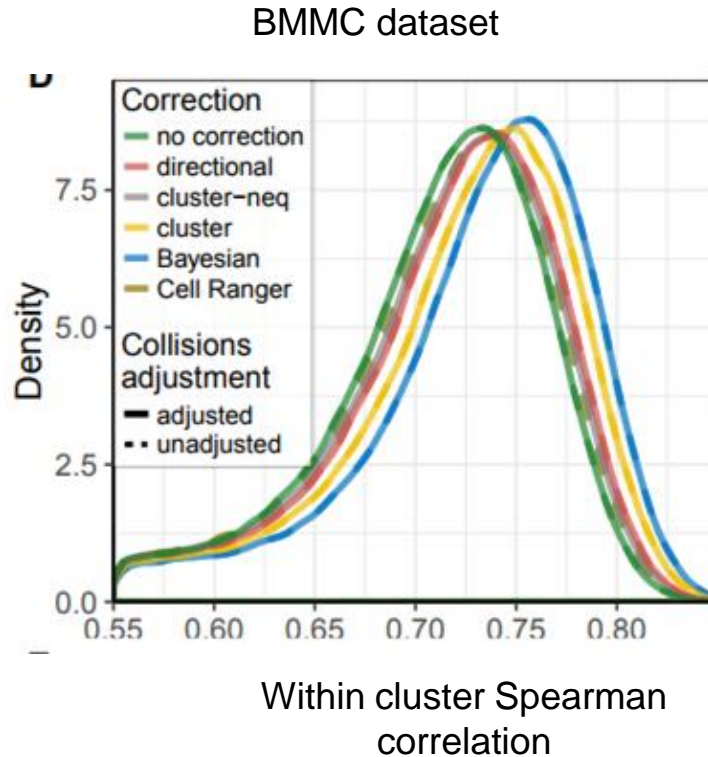
# Correcting For Sequencing Errors : UMI length



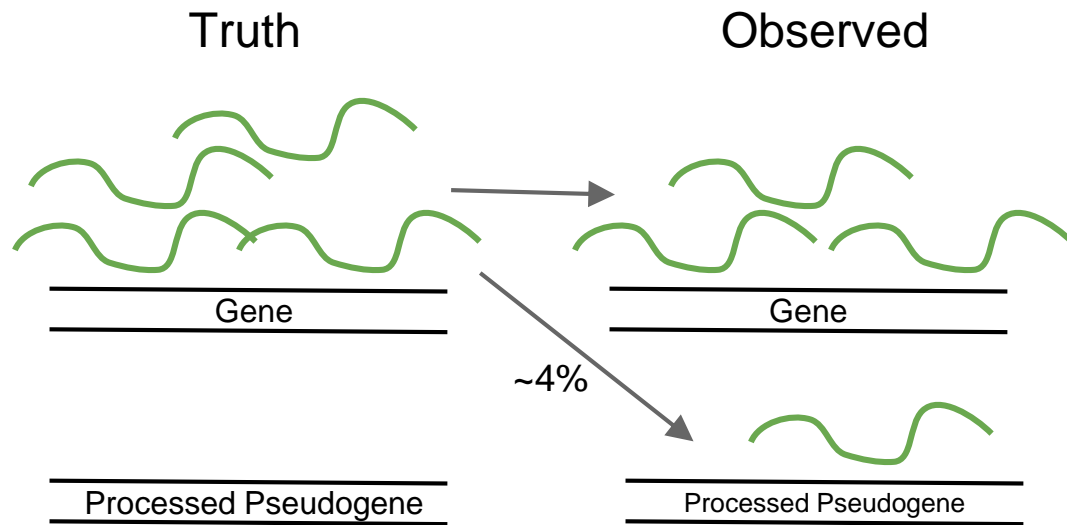
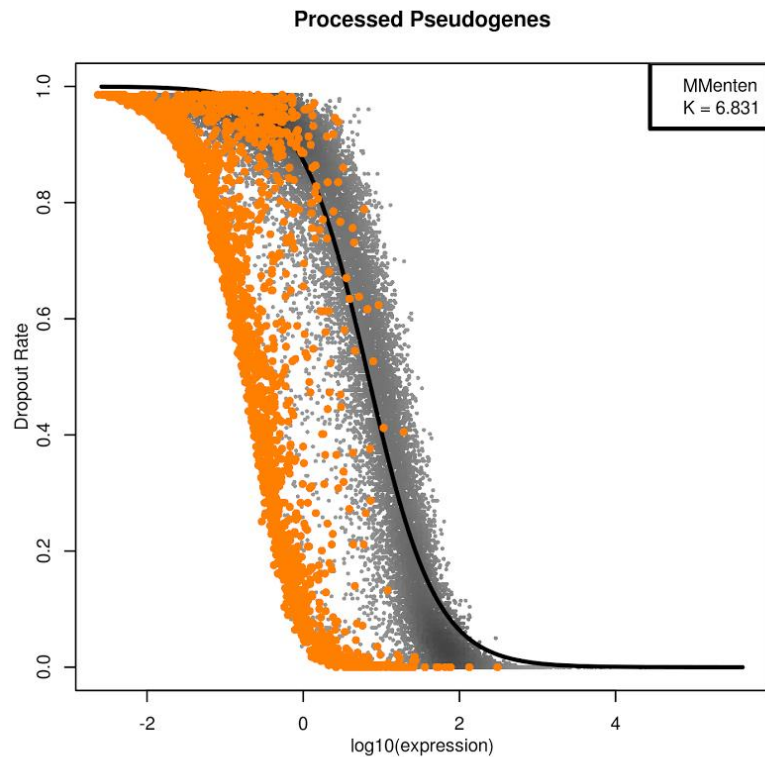
# dropEst: error correction + uneven UMIs

Bayesian Seq Error correction:

- UMI frequency
- number of adjacent UMIs
- empirical distribution of UMIs
- location/type of substitution



# Correcting For UMI Errors: Mismapping

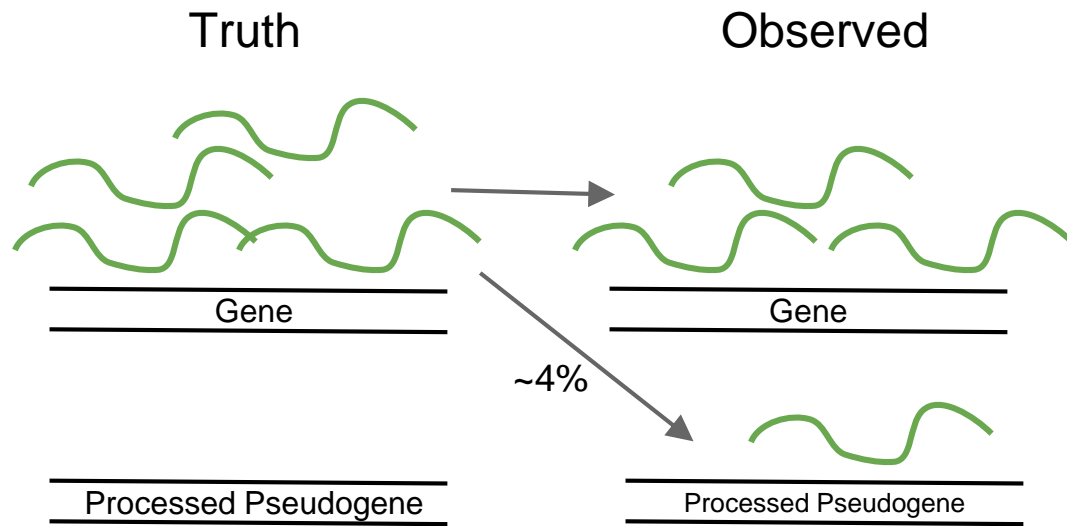


1% sequencing error rate x 100bp reads:  
4% of reads have 3+ sequencing errors

# Correcting For UMI Errors: Mismapping

No standard tools to correct but approaches to mitigate the effect:

- 1) Remove gene-UMI pairs observed in only one read.
- 2) Remove/mask processed pseudogenes and other non-polyA containing transcripts from the genomes/transcriptome
- 3) Remove multimapping reads prior to counting UMIs

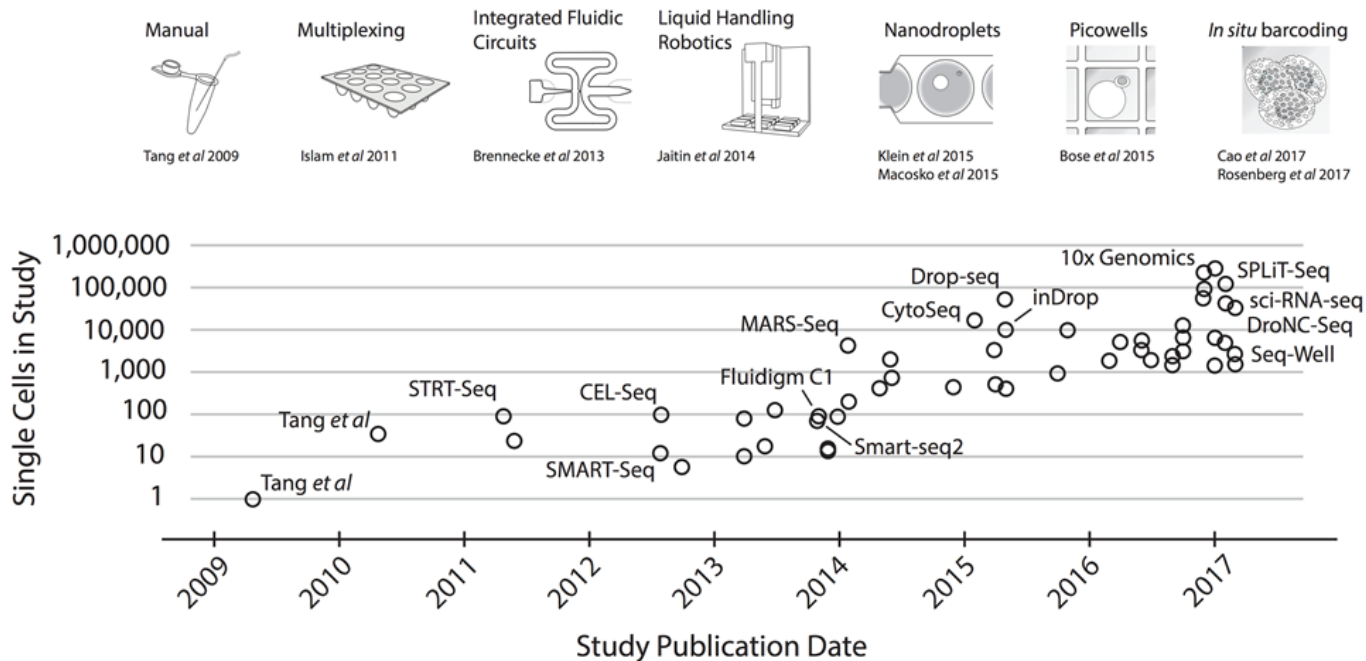


1% sequencing error rate x 100bp reads:  
4% of reads have 3+ sequencing errors



# UMIs - Realistic scRNASeq situations

UMIs are generally used with very high throughput methods.



# UMIs - Realistic scRNASeq situation

UMIs are generally used with very high throughput methods.

Can be very shallowly sequenced  $< 10,000$  umi/cell.

Many gene-UMI pairs will be supported by only a single read.

Most genes will have mean expression levels of  $< 20$  UMIs per cell.

Thus, all UMI sequencing error correction methods will be roughly equally sufficient and collision probabilities will generally be low.

In practice, there may be other factors that cause bigger biases/errors in our analysis.

Thus, very sophisticated corrections may not be worth the effort.

# Summary

Reads overlapping any genomic features can be quantified after mapping

- HT-Seq, featureCounts, Libinorm
- Excluding/down weighting multimapping reads will bias this quantification
- Filtering the genome for processed pseudogene, etc.. can reduce multimapping issues
- Libinorm correctly normalizes for protocol-specific gene coverage

Expression can be quantified directly using pseudo-aligners

- Differences between tools are small

Standard isoform quantification can be used on high-coverage full-transcript scRNASeq data

- Except eXpress
- But gene-level expression estimates are more accurate

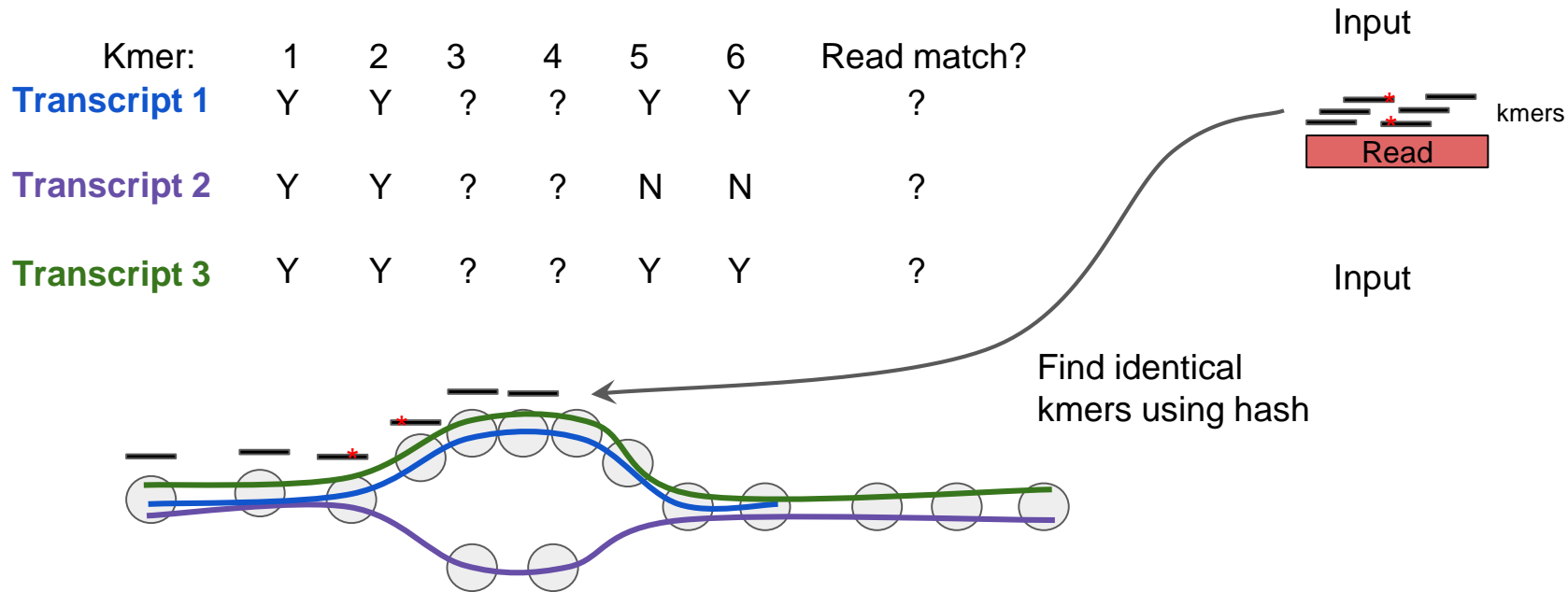
Quantifying intronic reads can be important for some cell-types

- Crucial for single-nuclei sequencing
- Dynamics : RNA Velocity

Unique molecular identifiers should be corrected for sequencing errors and collisions

- UMI-tools or dropEst
- Multi-mapping reads or UMIs with only 1 read support should be removed when counting UMIs

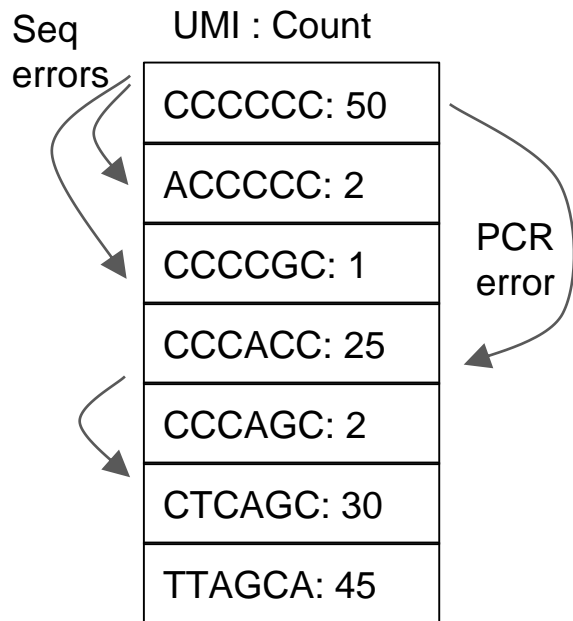
# How do Pseudo-Aligners quantify expression?



What happens in a read contains a sequencing error?

# Sequencing Errors (UMI-Tools)

## UMI Pool for Gene X

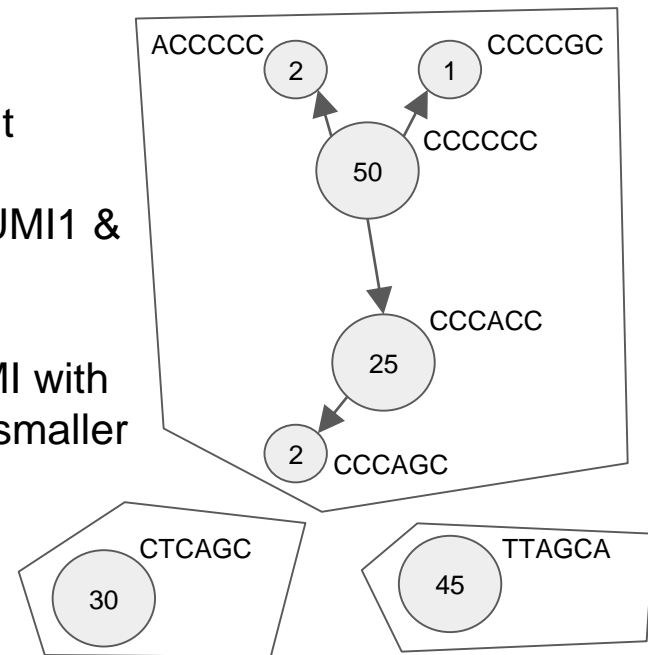


If:

- UMI1 and UMI2 have 1 nt different
- Difference in counts for UMI1 & UMI2  $\geq 2$

Then:

- Create an arrow from UMI with larger count to UMI with smaller count



Identify connected groups.

#Groups = Corrected UMI Count

Final count = 3

# Ribosomal protein expression drives clusters

Ribosomal proteins account for ~40% of UMIs in many 10X datasets.

