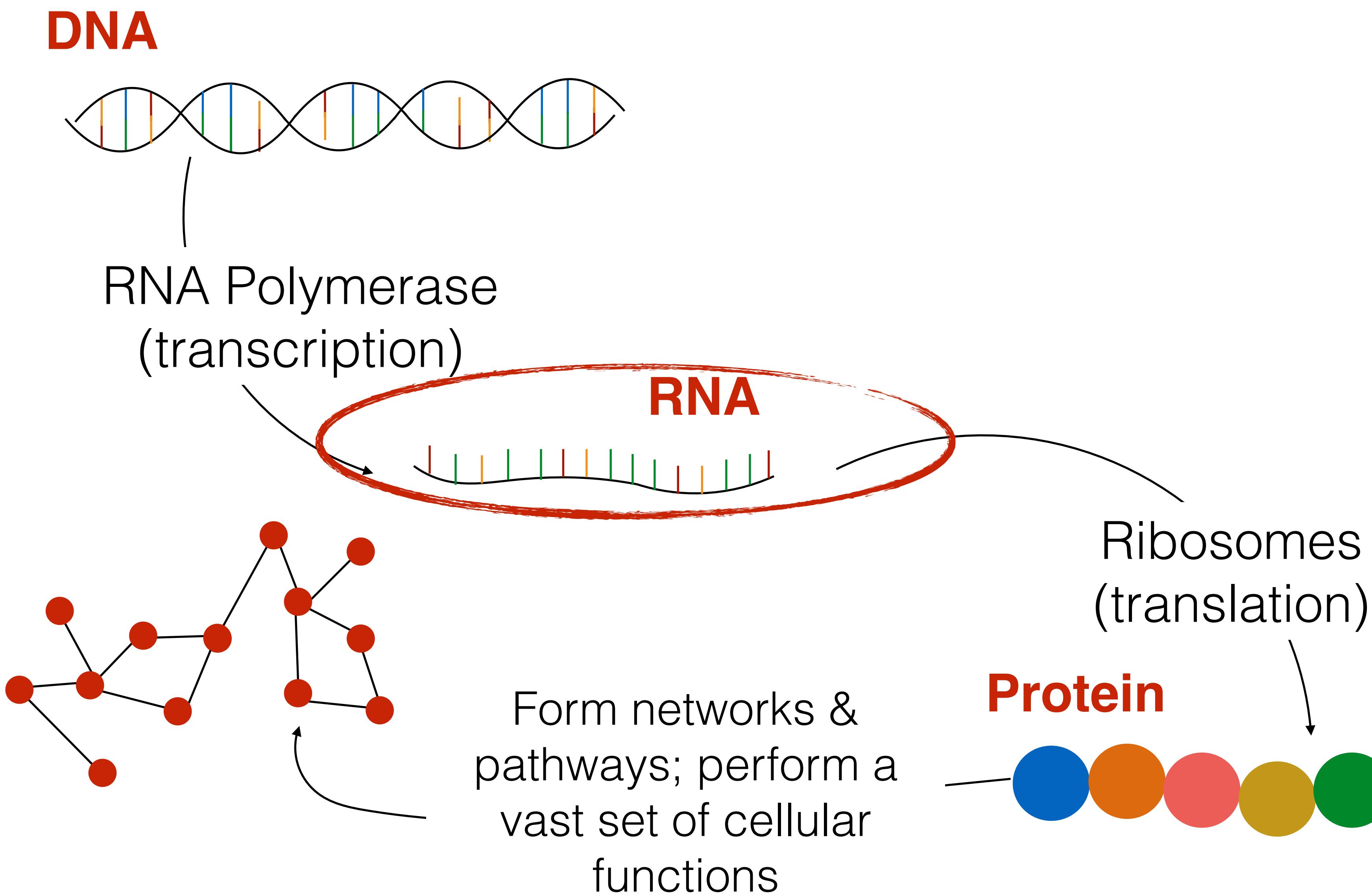


# Analyzing gene and transcript expression using RNA-seq

# “Flow” of information in the cell



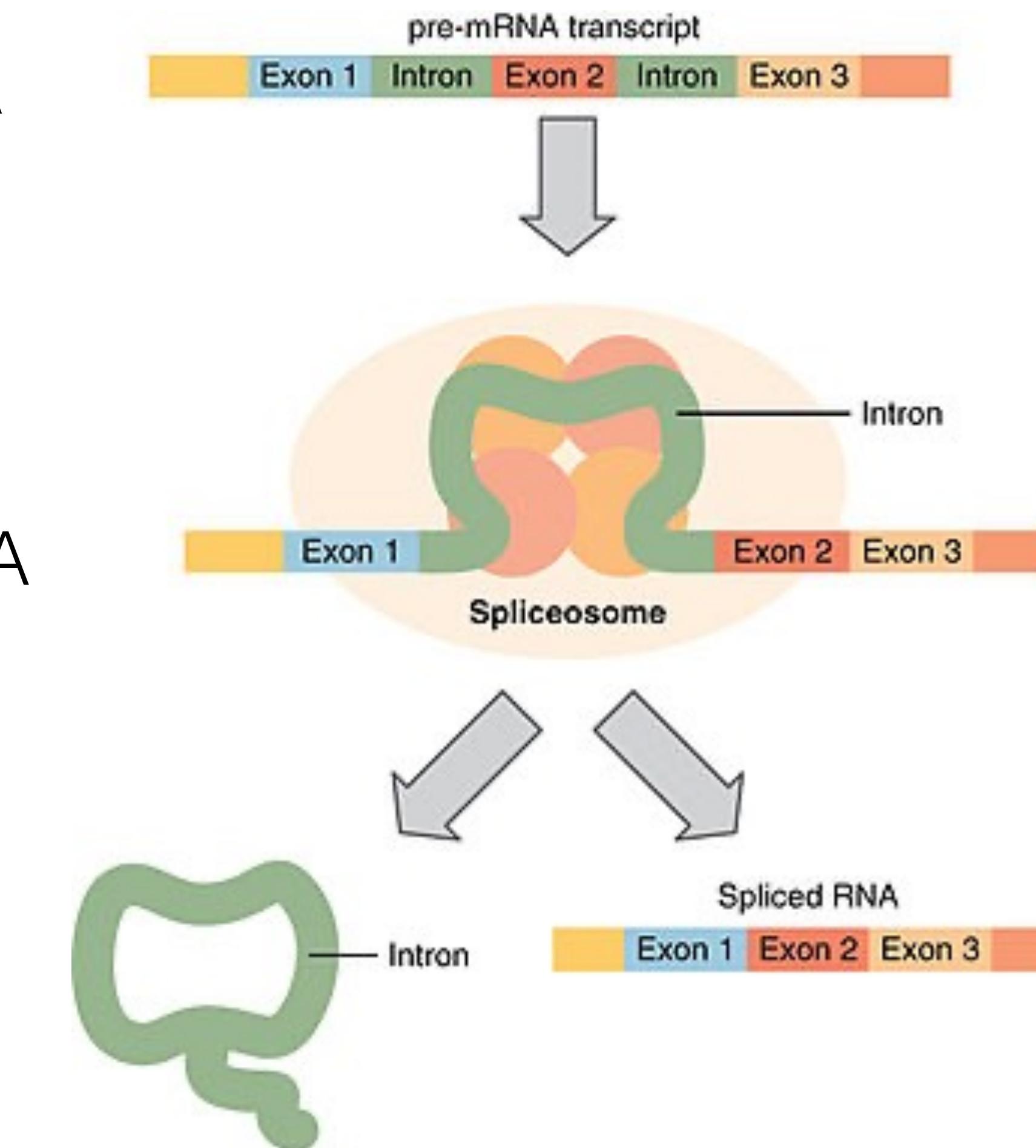
# RNA Splicing

DNA transcribed into pre-mRNA

Some “processing occurs”  
**capping & polyadenylation**

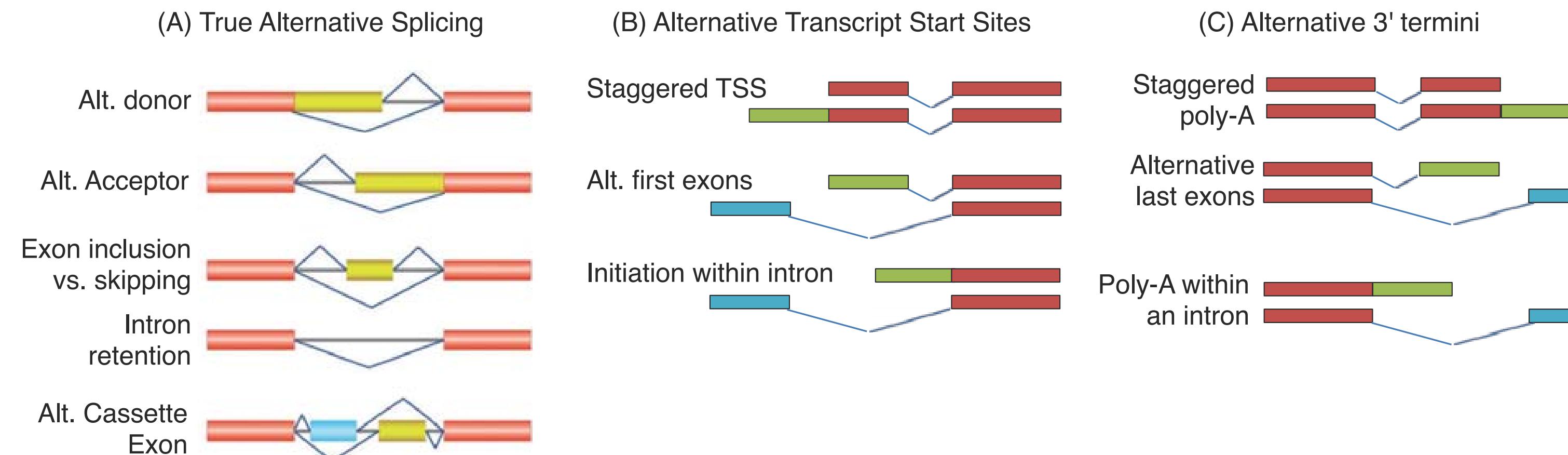
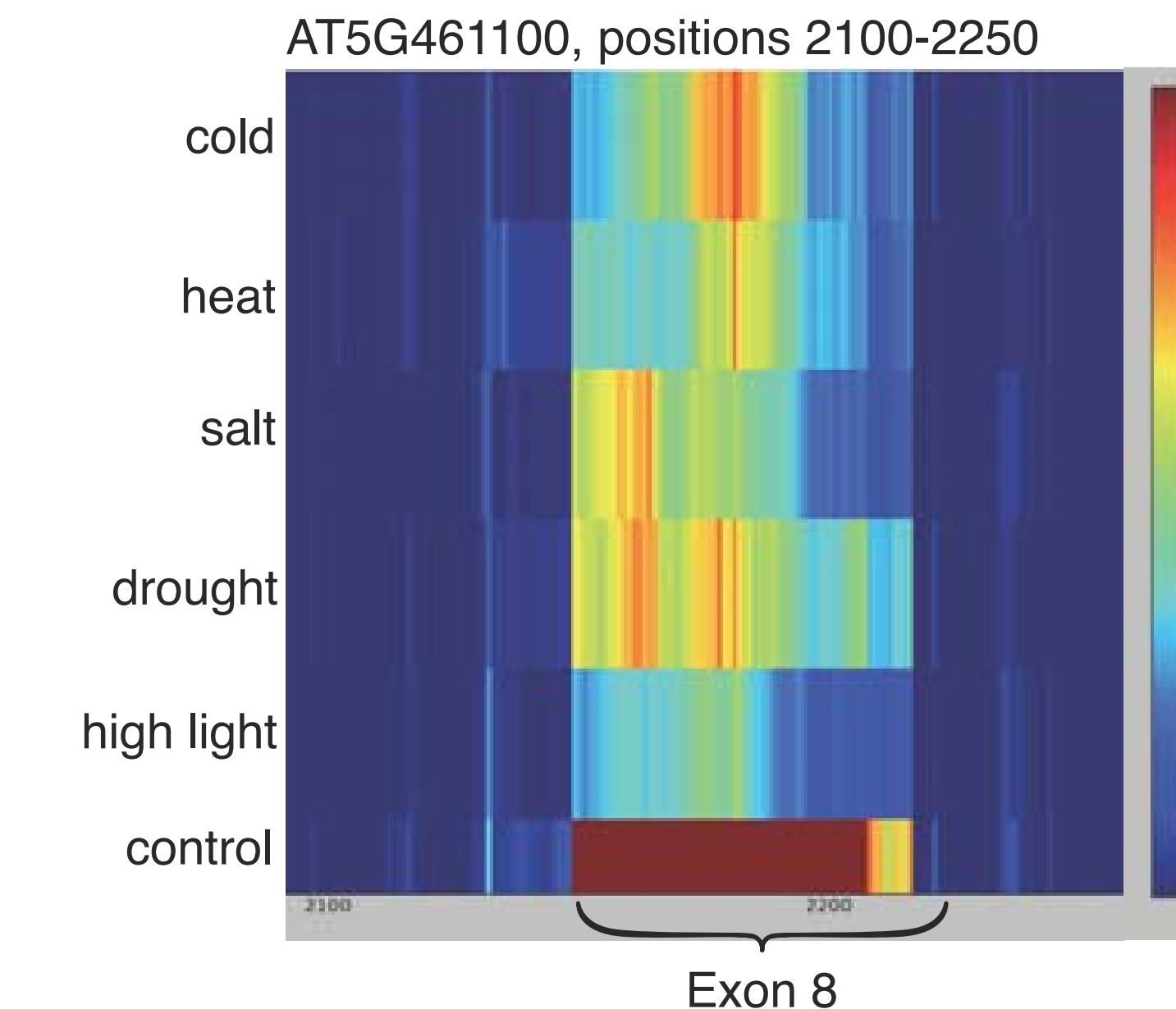
Introns removed from pre-mRNA

Introns removed resulting in  
*mature mRNA*

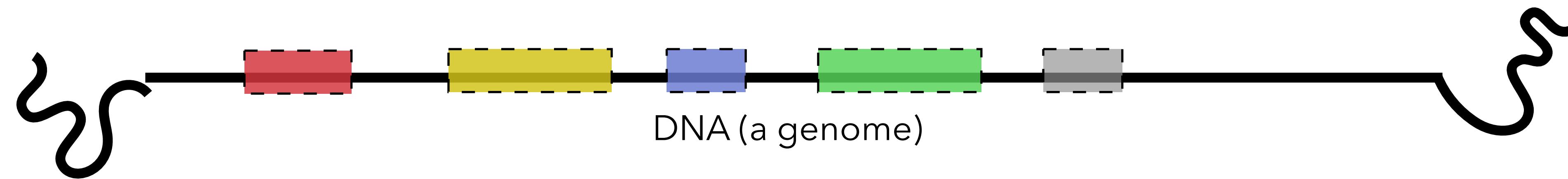


# Alternative Splicing & Isoform Expression

- Expression of genes can be measured via RNA-seq (sequencing transcripts)
- Sequencing gives you short (35-300bp length reads)

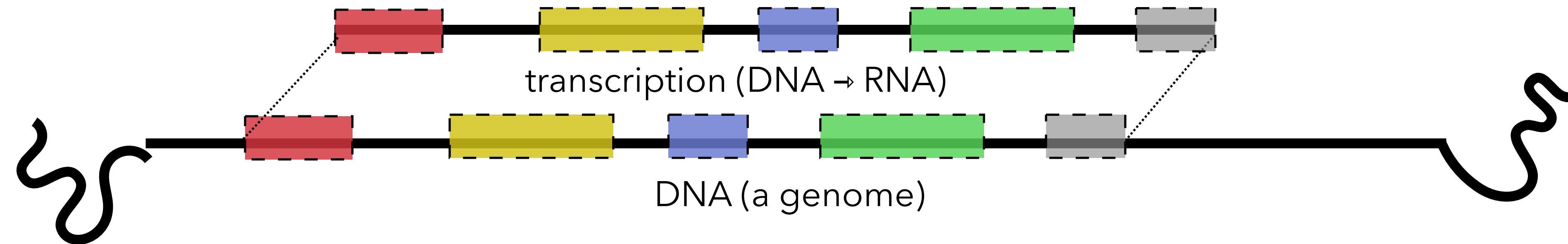


# What is RNA sequencing



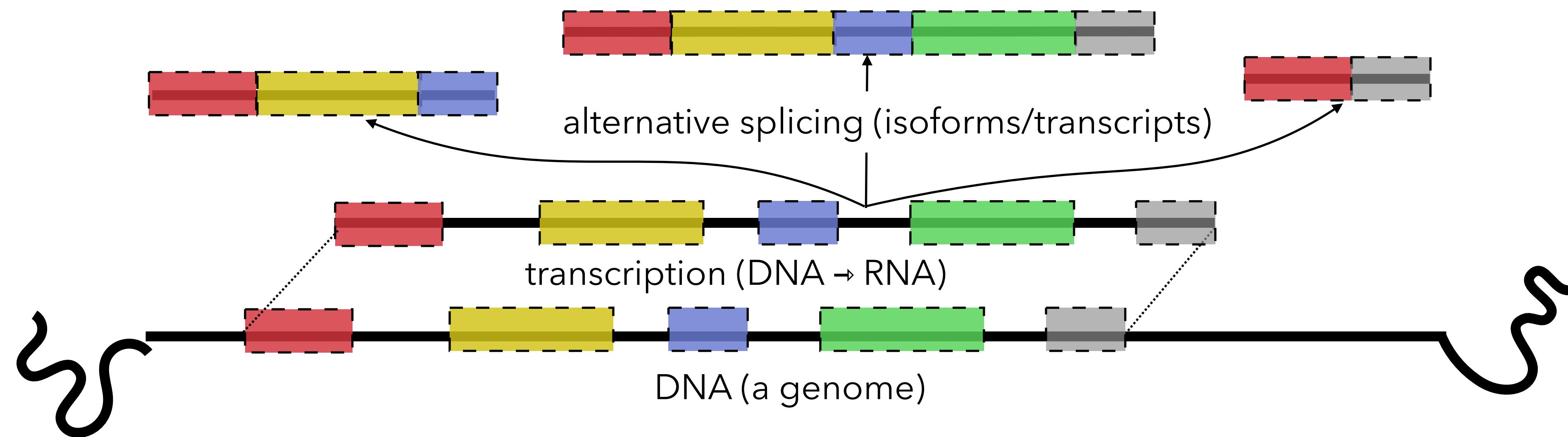
\* most protocols actually sequence complementary DNA (cDNA), not RNA directly

# What is RNA sequencing



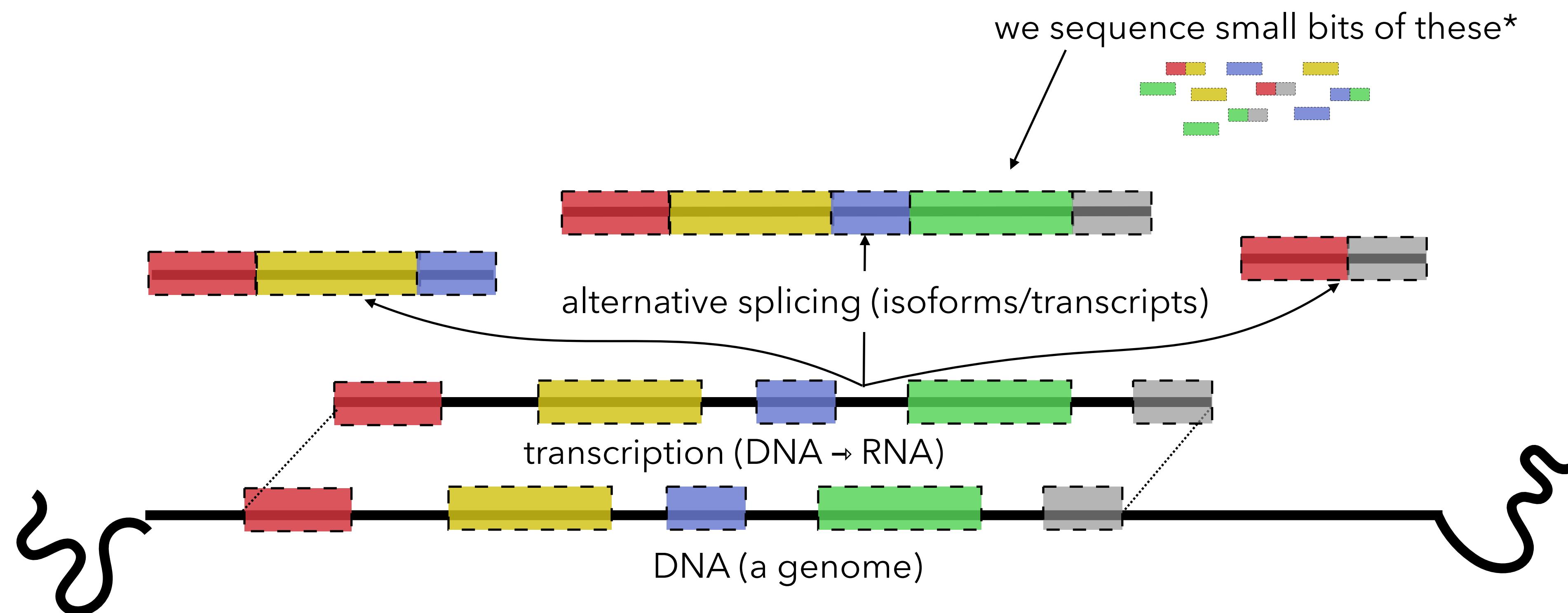
\* most protocols actually sequence complementary DNA (cDNA), not RNA directly

# What is RNA sequencing



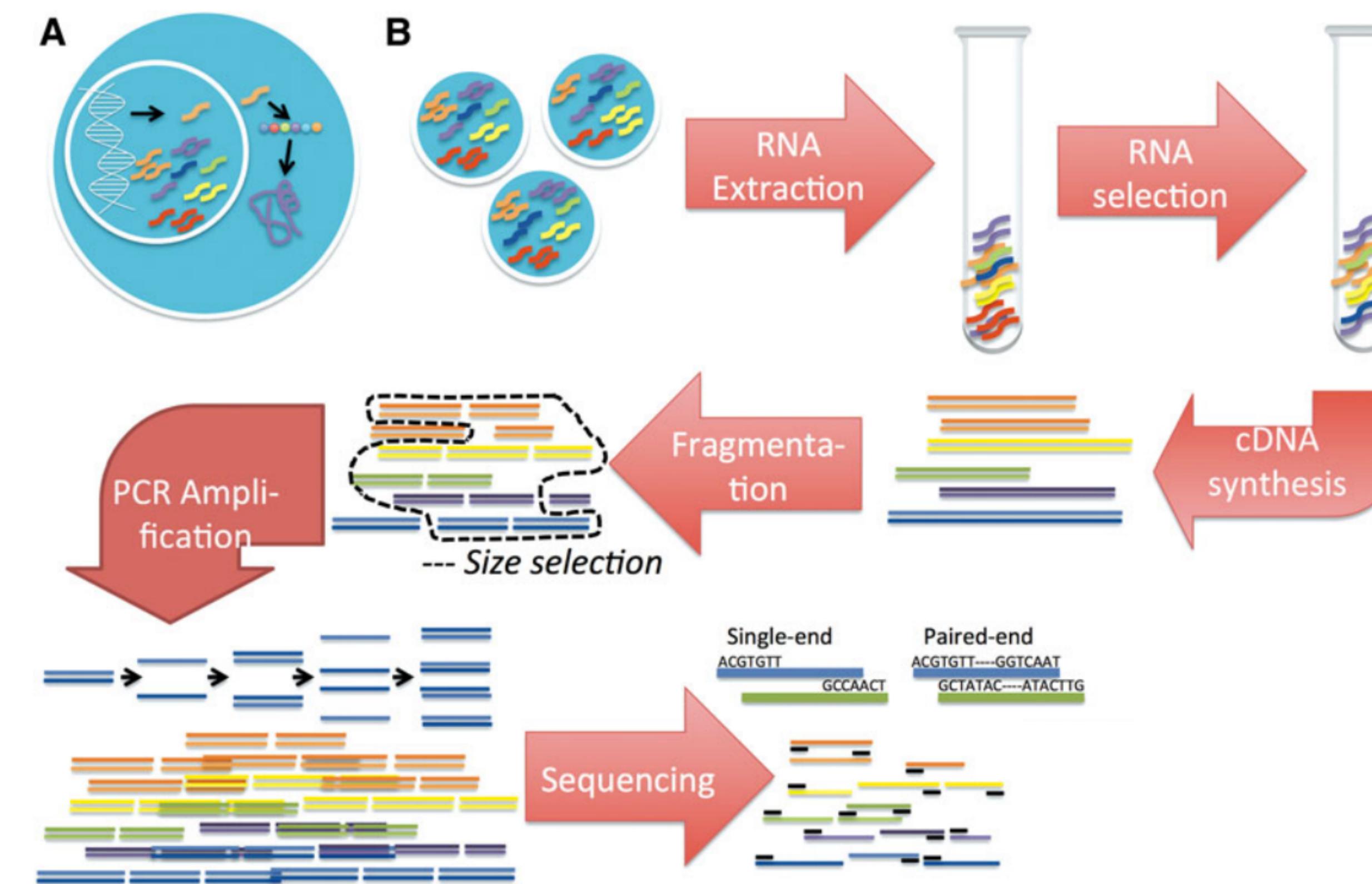
\* most protocols actually sequence complementary DNA (cDNA), not RNA directly

# What is RNA sequencing

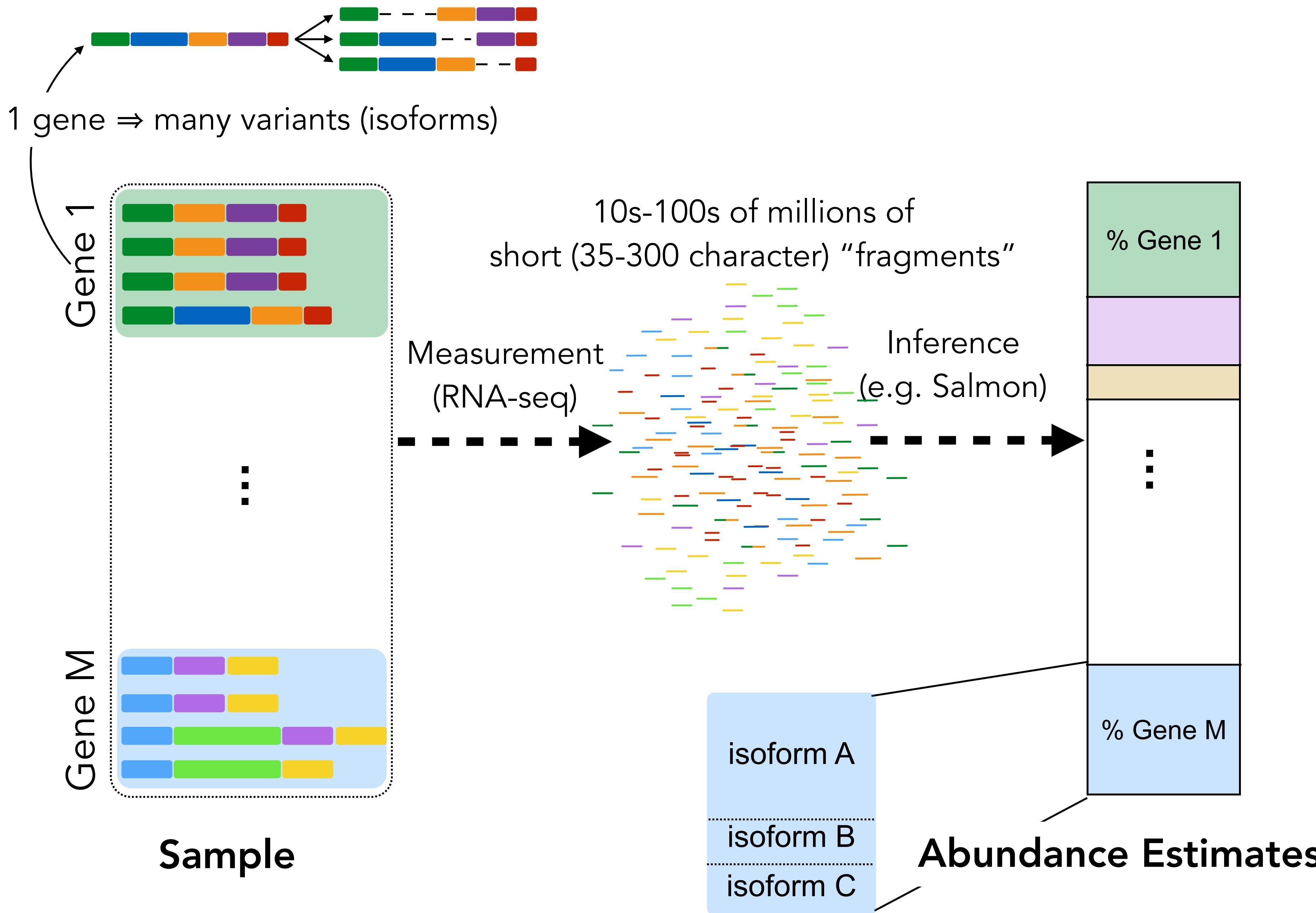


\* most protocols actually sequence complementary DNA (cDNA), not RNA directly

# Actual protocols are much more involved



# Transcript Quantification: An Overview

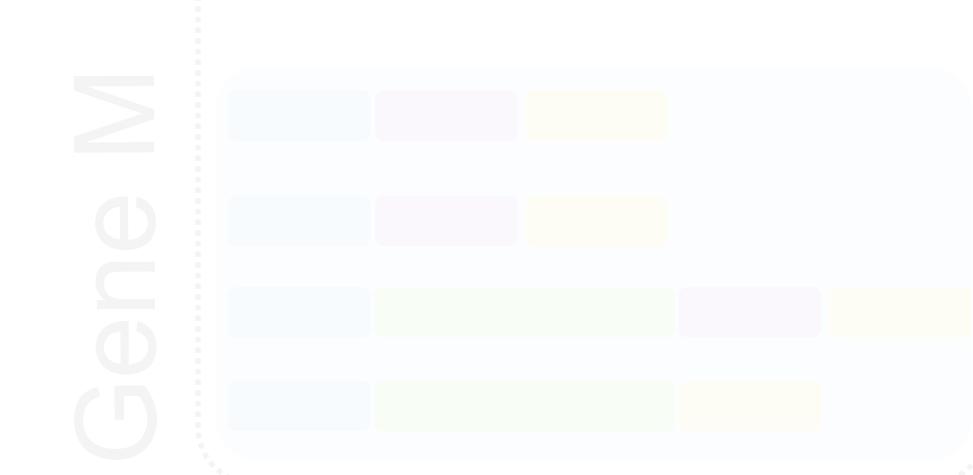




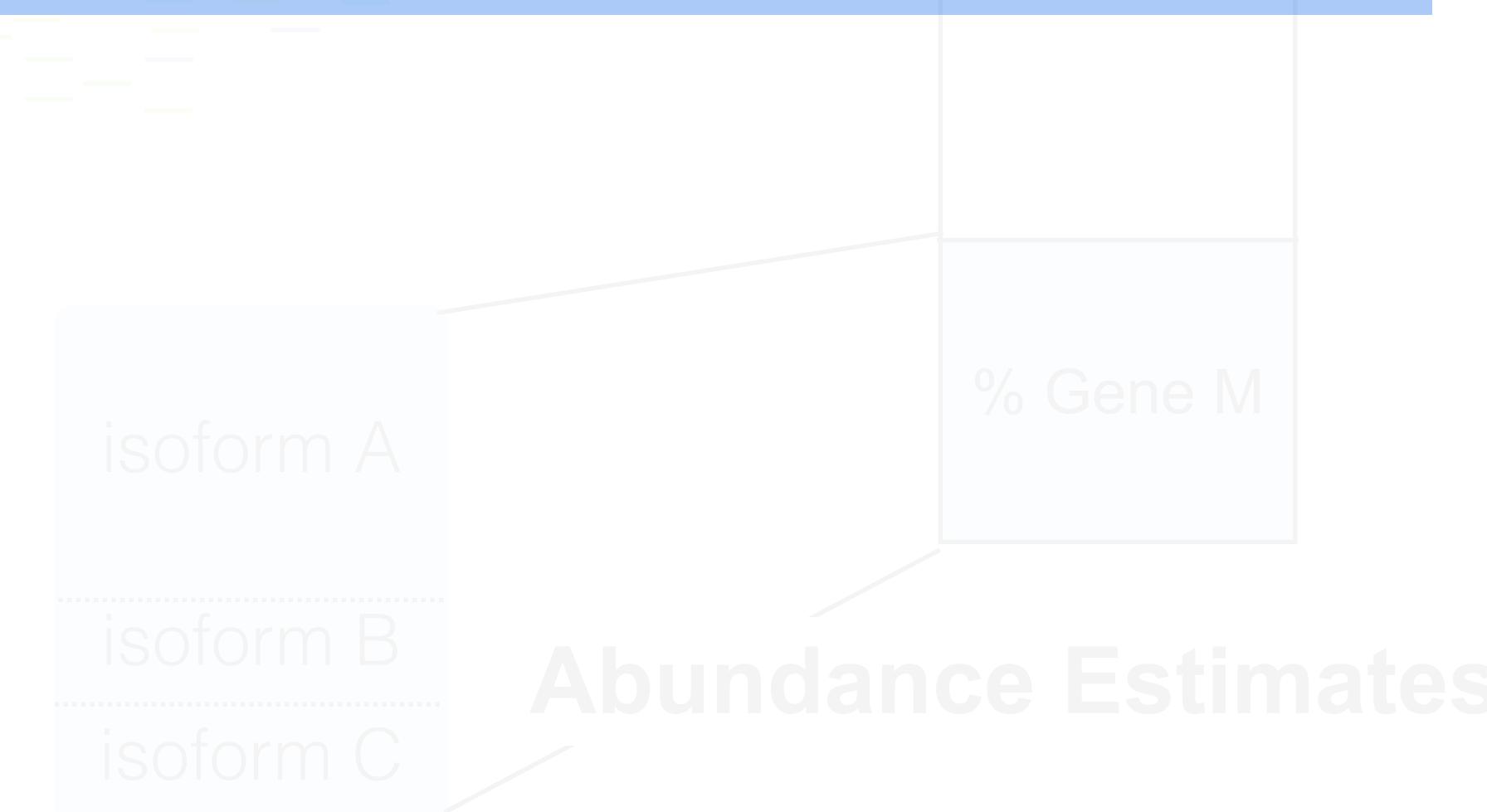
10s-100s of millions of  
short (35-300 character) “reads”

**Given:** (1) Collection of RNA-Seq fragments  
(2) A set of known (or assembled) transcript sequences

**Estimate:** The relative abundance of each transcript



Sample



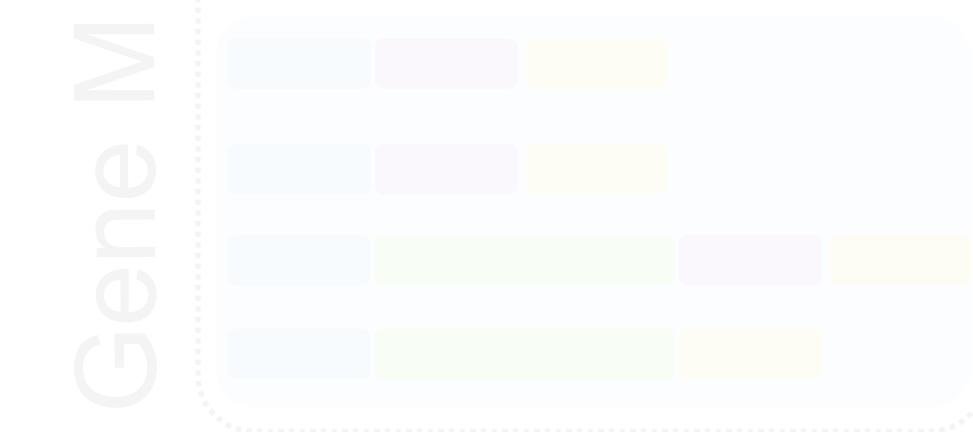


1 gene  $\Rightarrow$  many variants (isoforms)

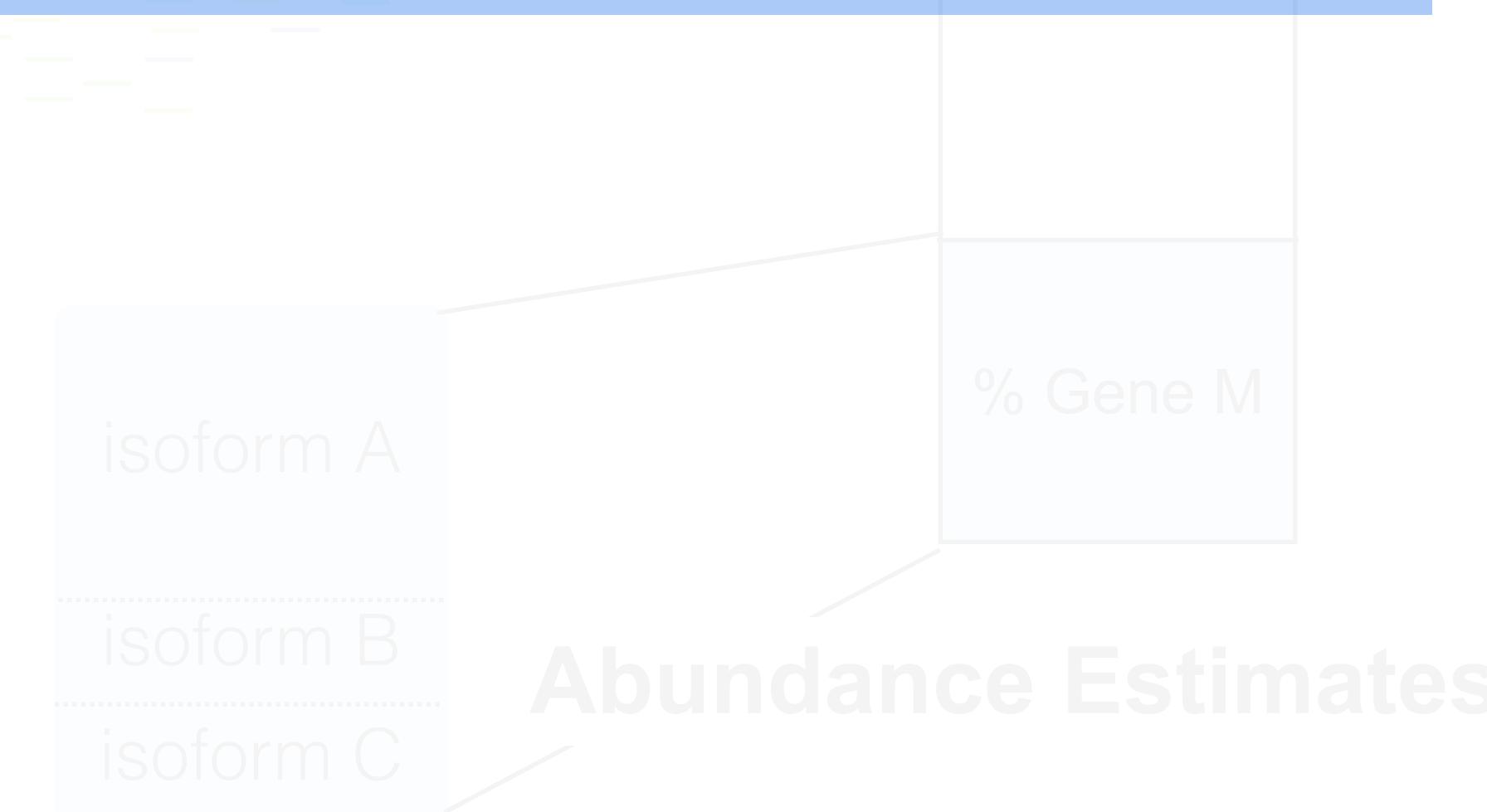
10s-100s of millions of  
short (35-300 character) “reads”

**Given:** (1) Collection of RNA-Seq fragments  
(2) A set of **known** (or assembled) transcript sequences

**Estimate:** The relative abundance of each transcript



Sample



Abundance Estimates

# Why not simply “count” reads

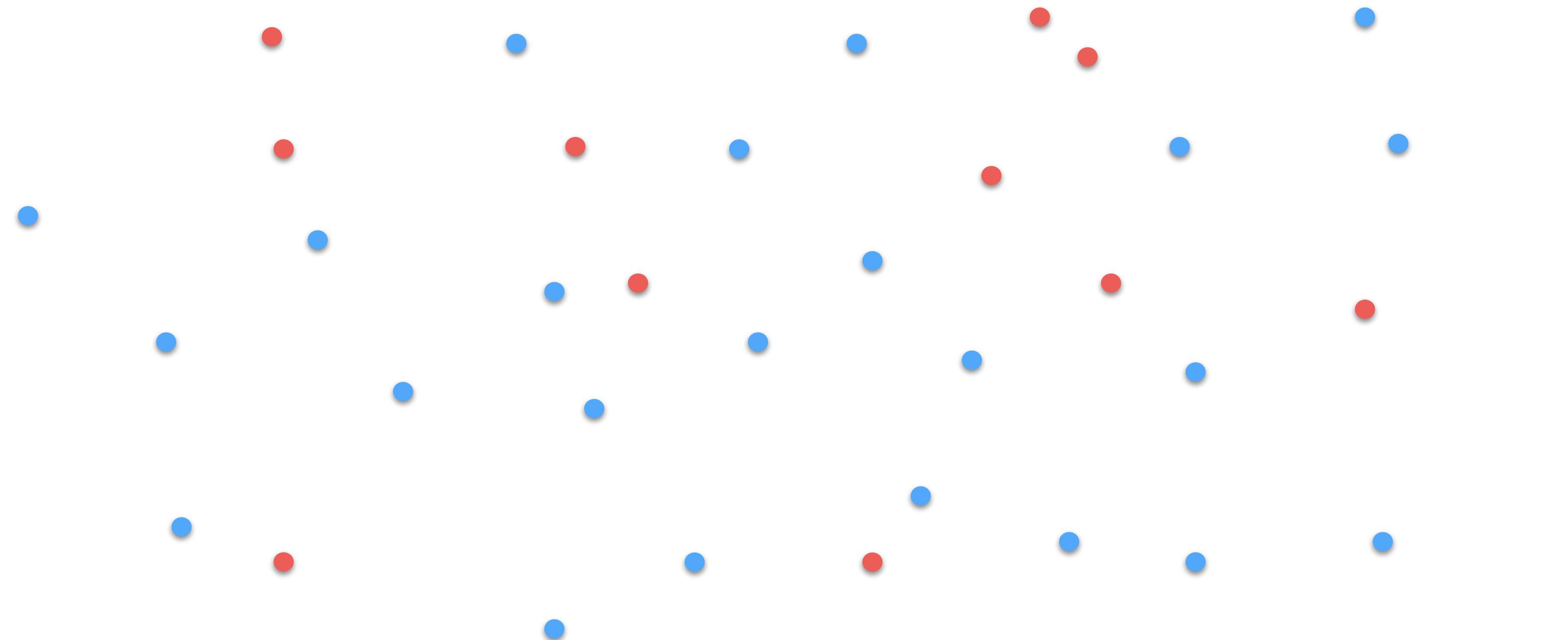
The RNA-seq reads are drawn from transcripts, and our (spliced) aligners let us map them back to the transcripts on the genome from which they originate.

Problem: How do you handle reads that align equally-well to multiple isoforms / or multiple genes?

- Discarding multi-mapping reads leads to incorrect and biased quantification
- Even at the gene-level, the transcriptional output of a gene should depend on what isoforms it is expressing.

# First, consider this non-Biological example

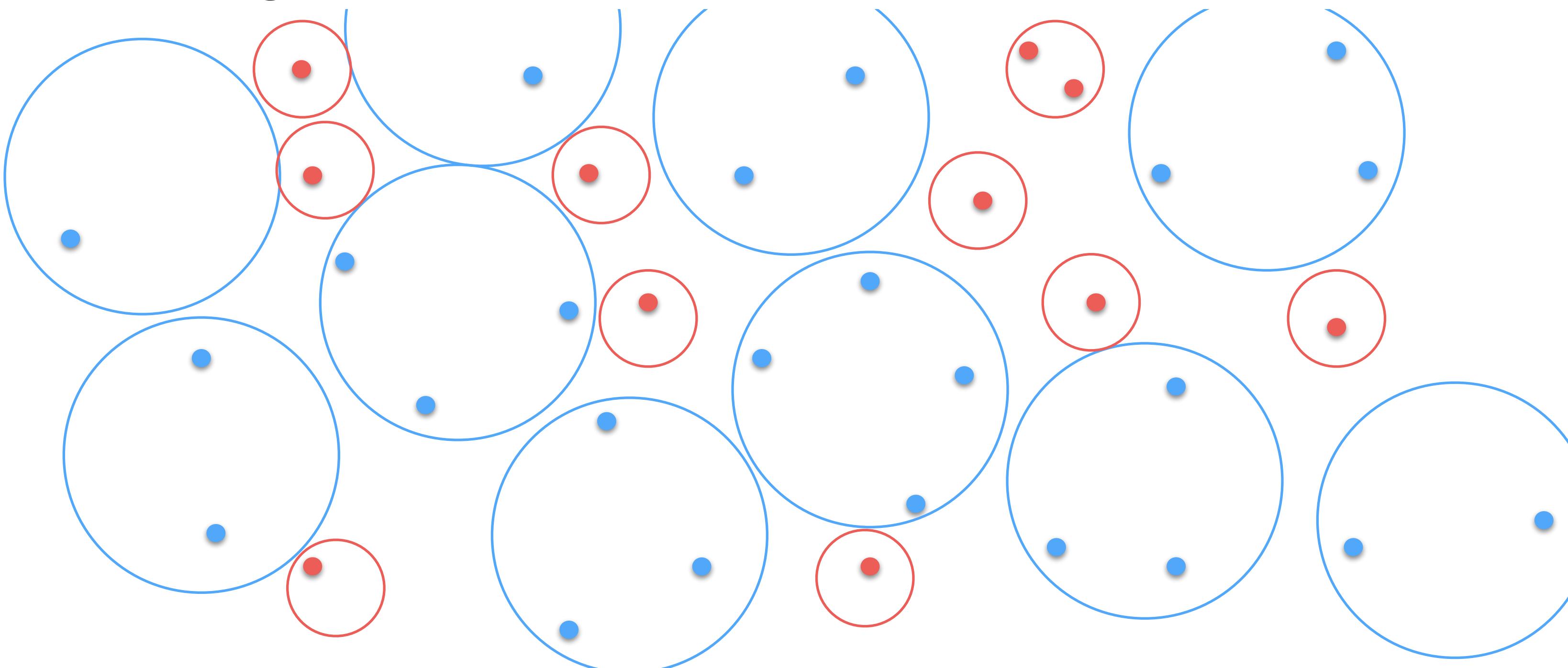
Imagine I have two colors of circle, **red** and **blue**. I want to estimate the **fraction of circles** that are **red** and **blue**. I'll *sample* from them by tossing down darts.



Here, a dot of a color means I hit a circle of that color.  
What type of circle is more prevalent?  
What is the fraction of red / blue circles?

# First, consider this non-Biological example

Imagine I have two colors of circle, **red** and **blue**. I want to estimate the **fraction of circles** that are **red** and **blue**. I'll *sample* from them by tossing down darts.



You're missing a **crucial piece of information!**  
**The areas!**

# First, consider this non-Biological example

Imagine I have two colors of circle, **red** and **blue**. I want to estimate the **fraction of circles** that are **red** and **blue**. I'll *sample* from them by tossing down darts.

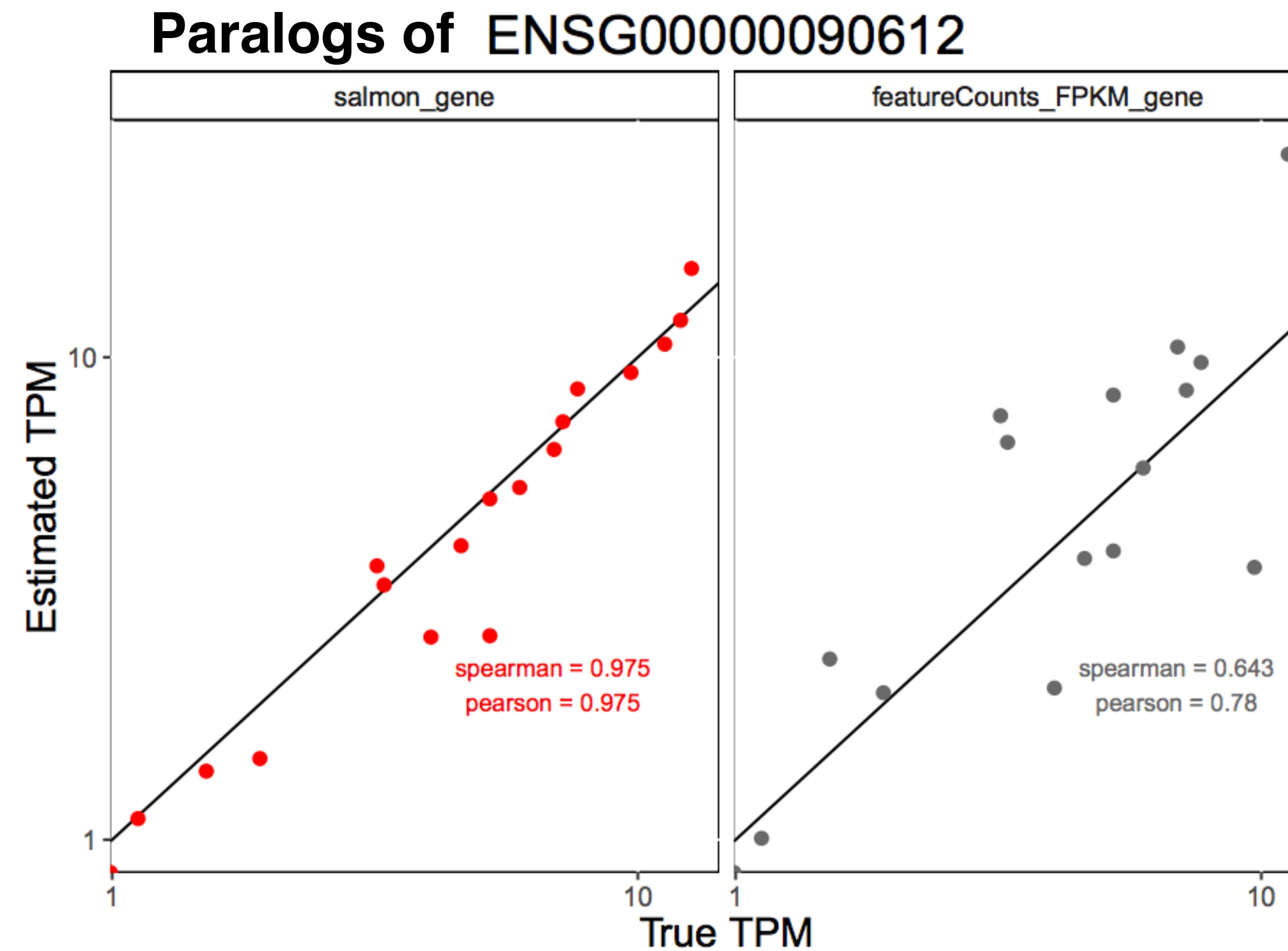
You're missing a **crucial piece of information!**

**The areas!**

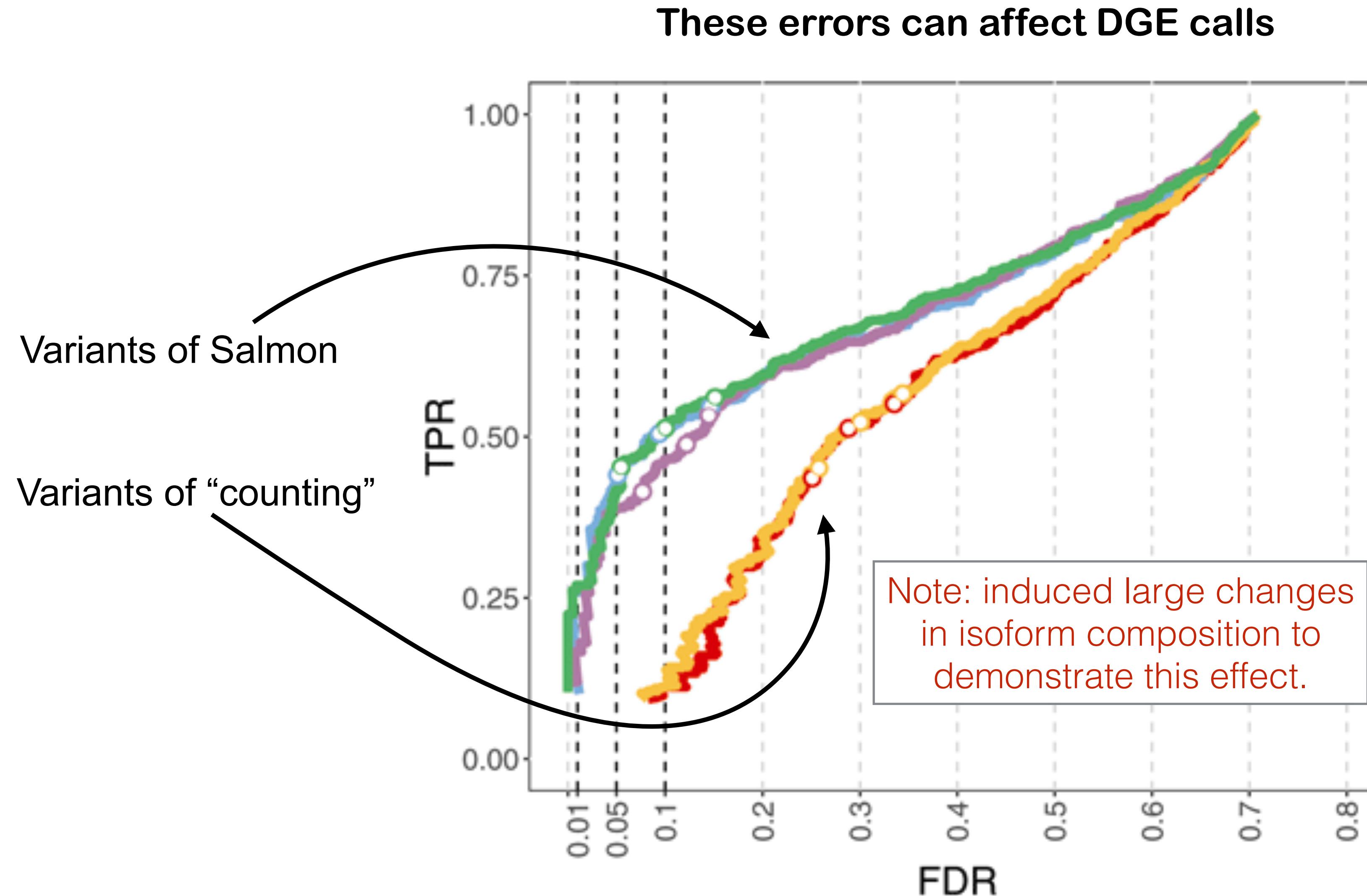
There is an analog in RNA-seq, one needs to know the **length** of the target from which one is drawing to meaningfully assess abundance!

# Resolving multi-mapping is fundamental to quantification

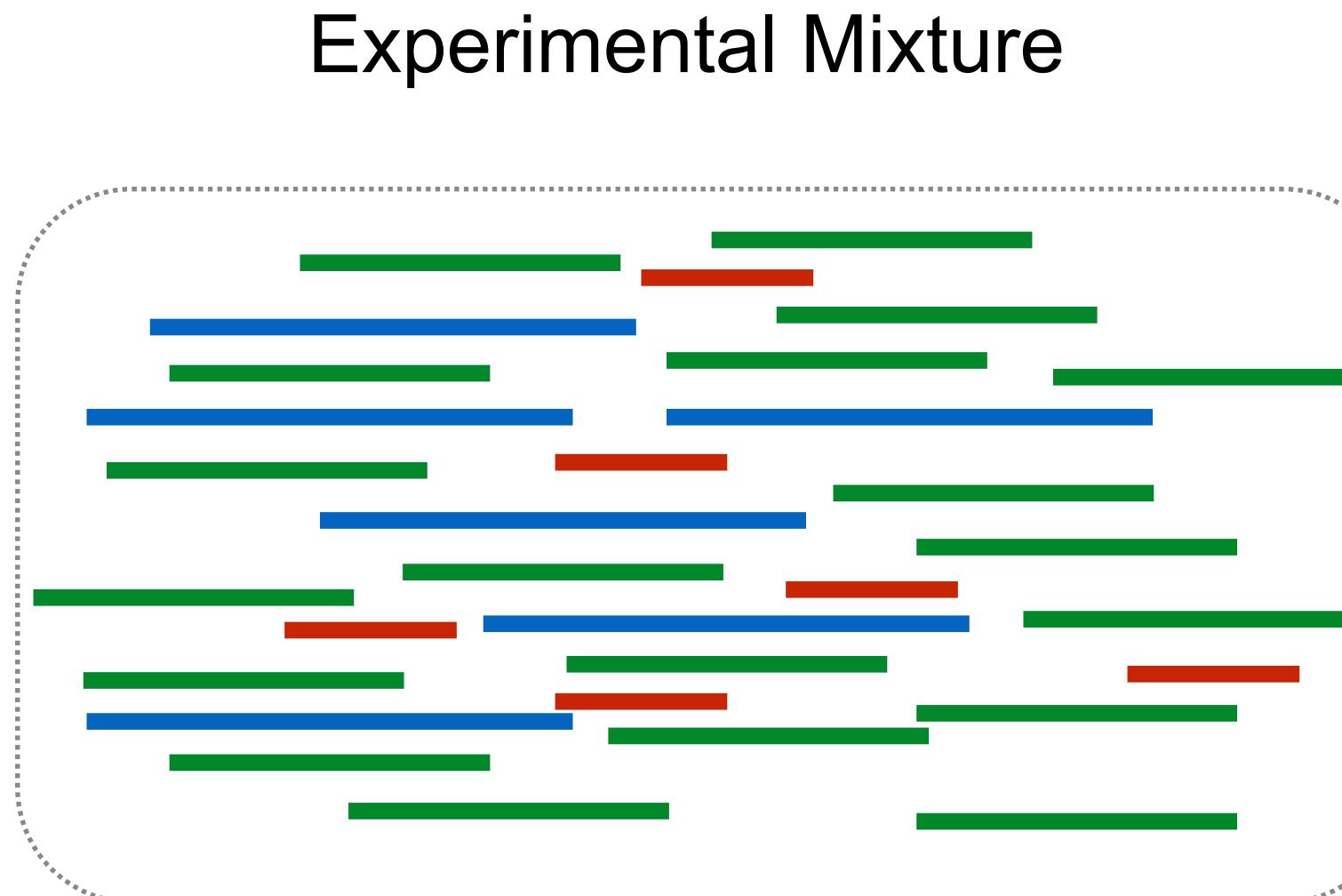
Can even affect abundance estimation in **absence** of alternative-splicing  
(e.g. paralogous genes)



# Resolving multi-mapping is fundamental to quantification



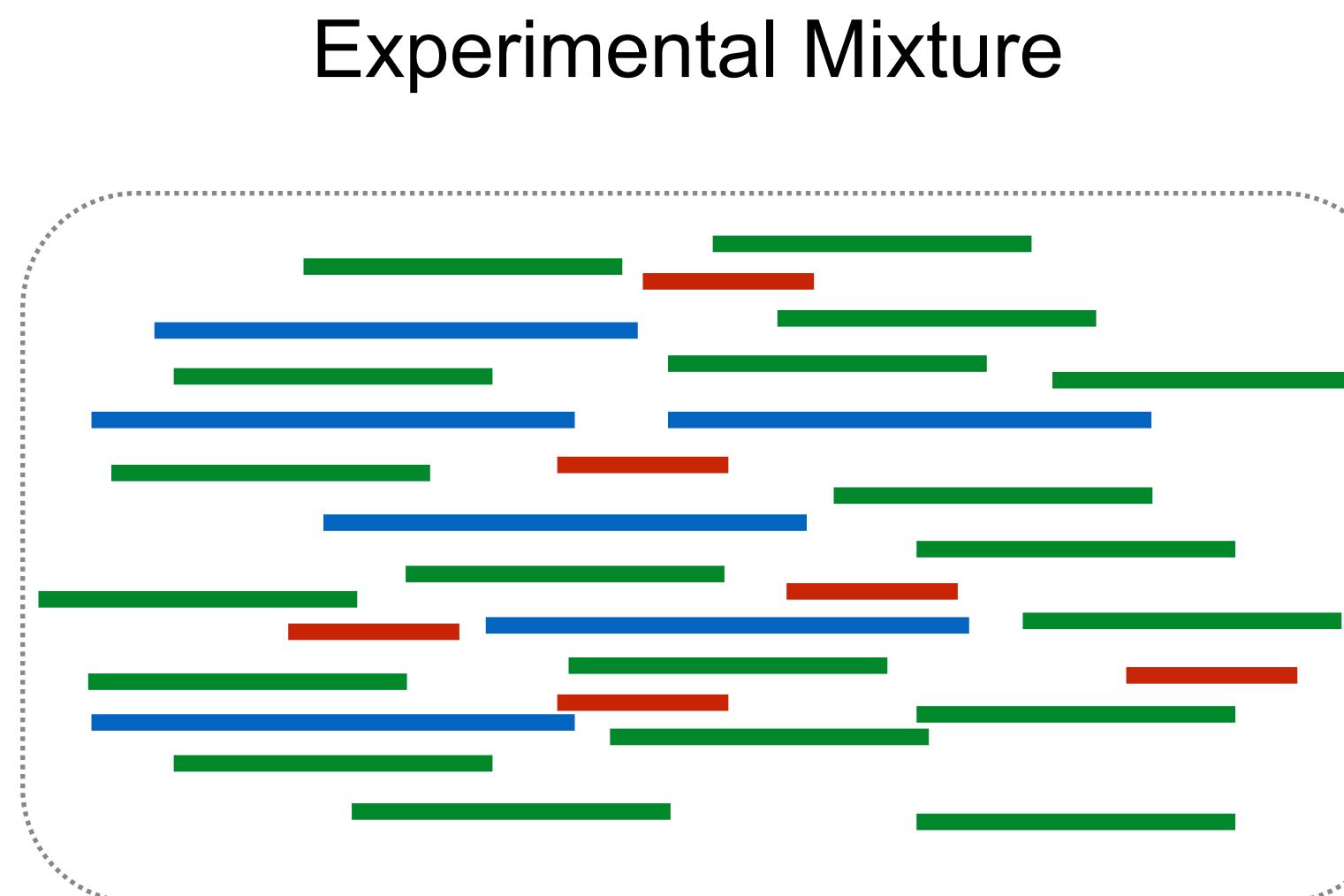
# How can we perform inference from sequenced fragments?



In an unbiased experiment,  
sampling fragments depends on:

- # of copies of each txp type
- length of each txp type

# How can we perform inference from sequenced fragments?

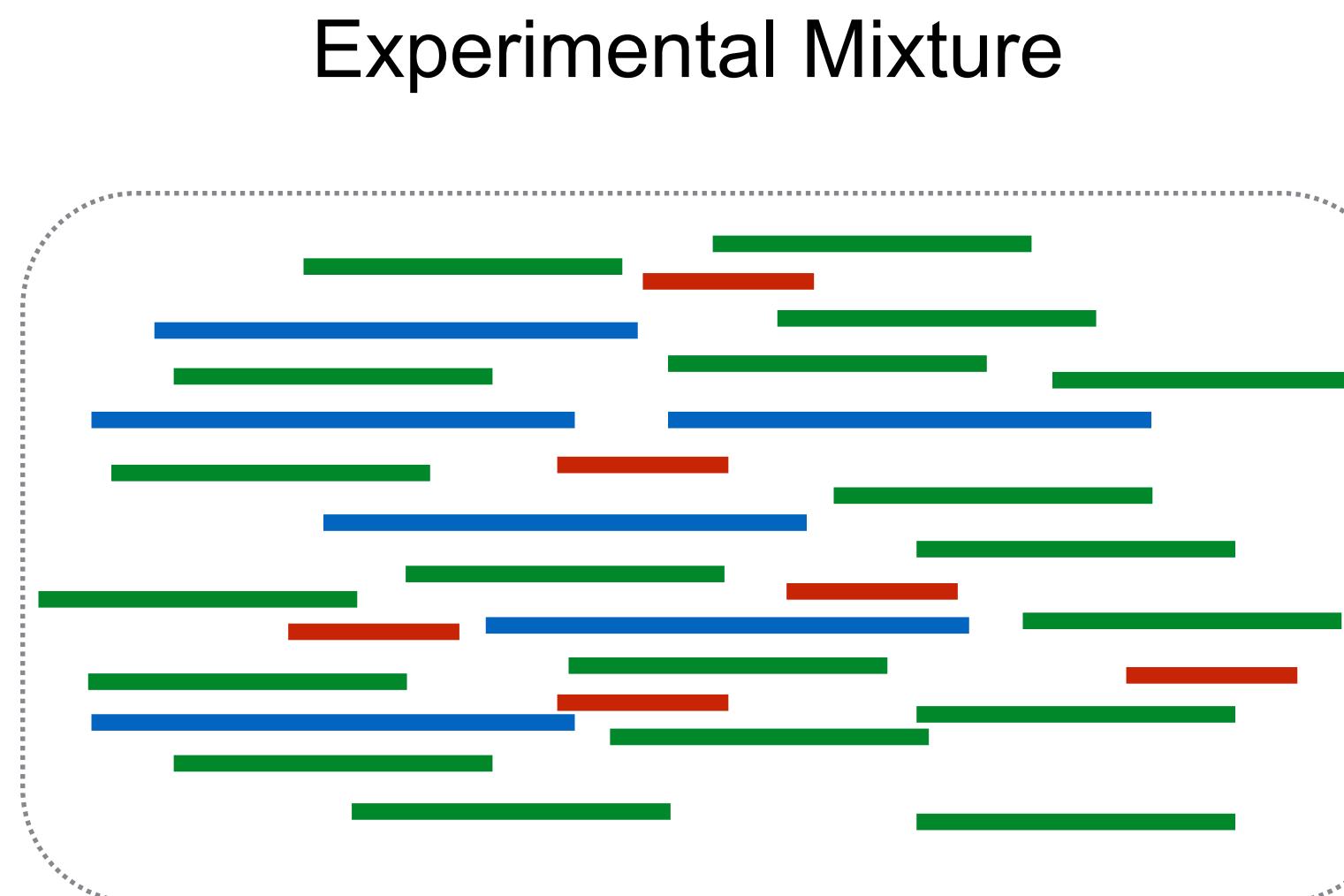


$\text{length}(\text{---}) = 100$

In an unbiased experiment,  
sampling fragments depends on:

- # of copies of each txp type
- length of each txp type

# How can we perform inference from sequenced fragments?

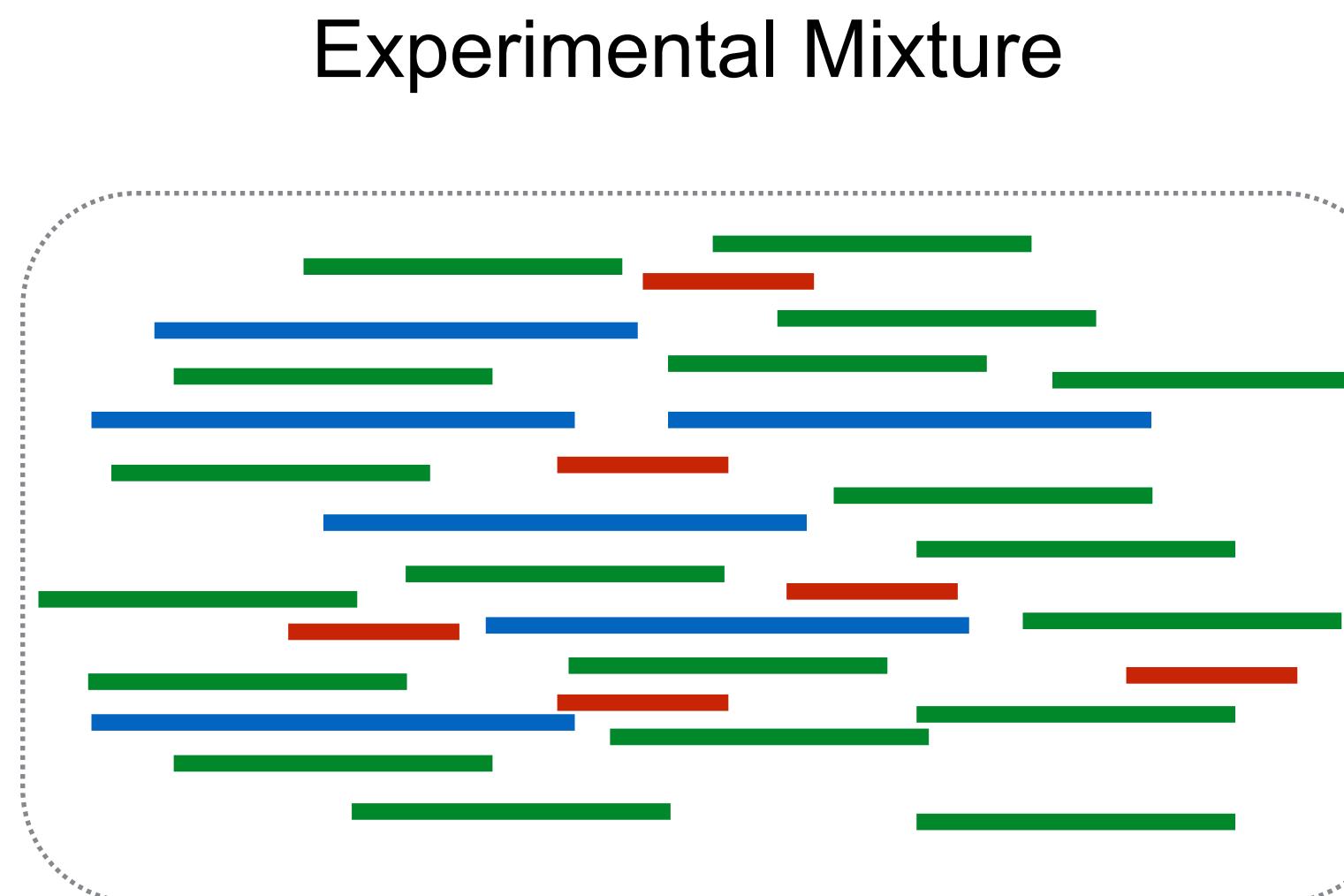


length(—) = 100 x 6 copies

In an unbiased experiment,  
sampling fragments depends on:

- # of copies of each txp type
- length of each txp type

# How can we perform inference from sequenced fragments?

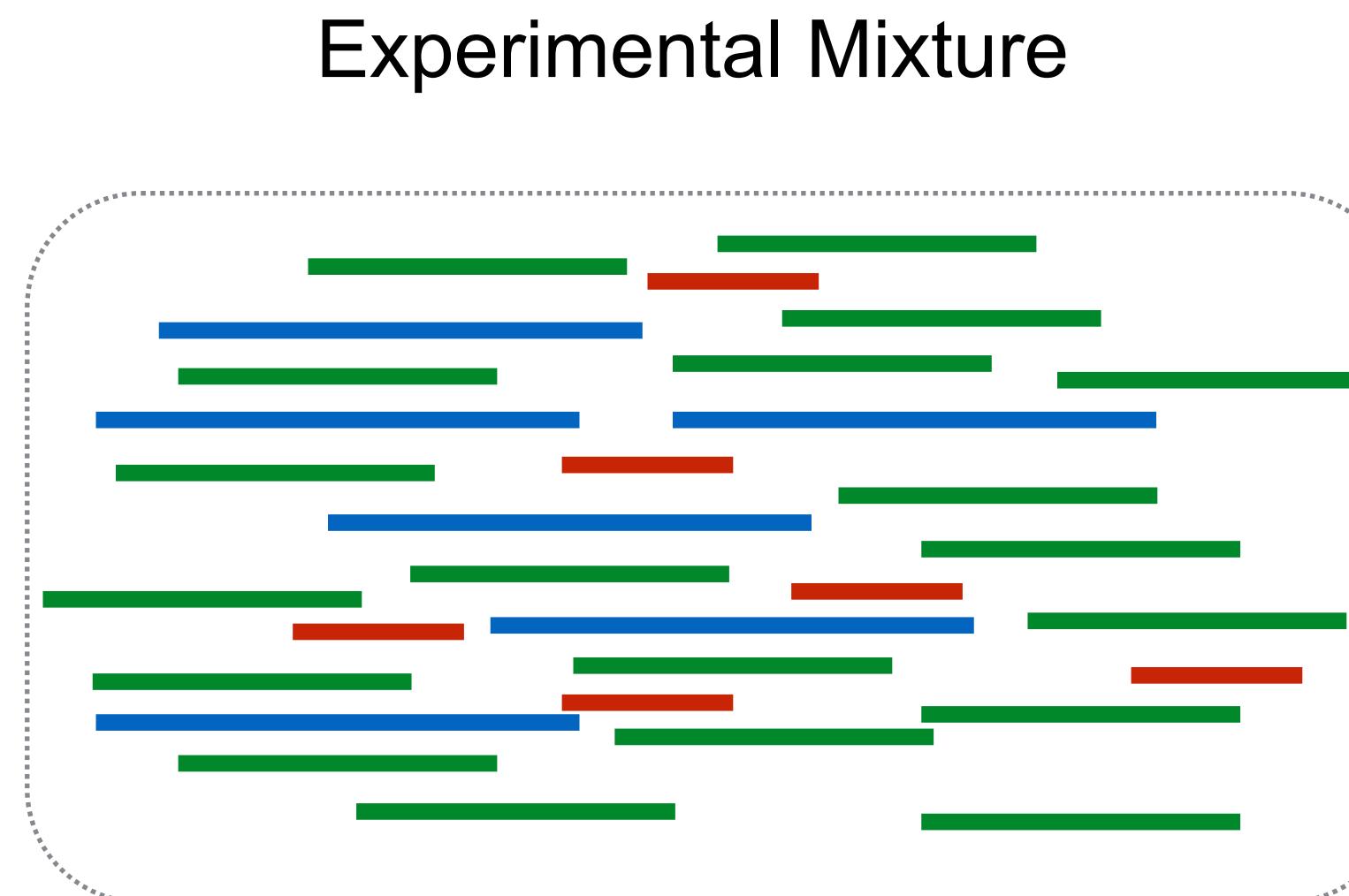


length( ) = 100 x 6 copies = 600 nt

In an unbiased experiment,  
sampling fragments depends on:

- # of copies of each txp type
- length of each txp type

# How can we perform inference from sequenced fragments?



In an unbiased experiment,  
sampling fragments depends on:

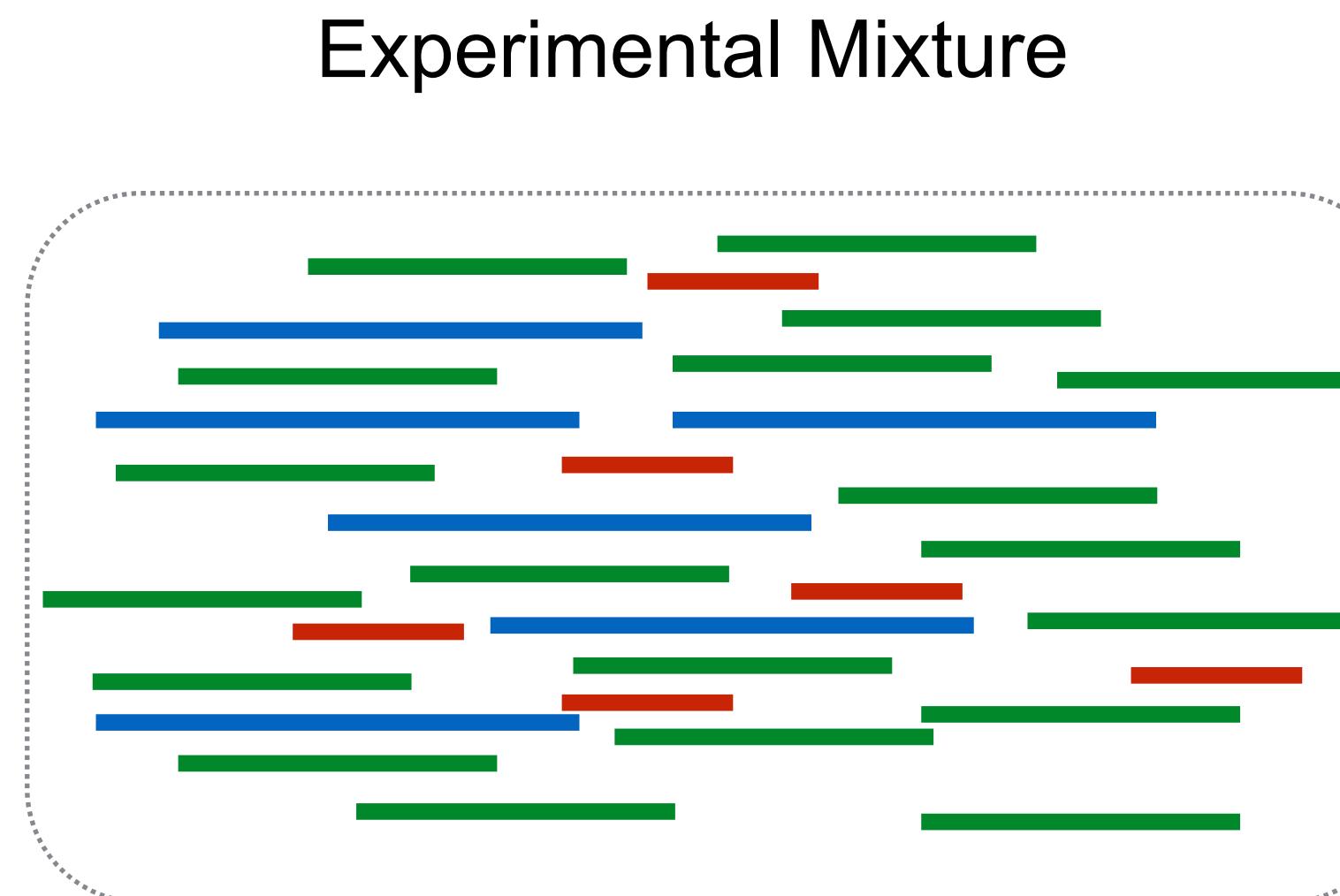
- # of copies of each txp type
- length of each txp type

$$\text{length}(\text{---}) = 100 \times 6 \text{ copies} = 600 \text{ nt}$$

$$\text{length}(\text{---}) = 66 \times 19 \text{ copies} = 1254 \text{ nt}$$

$$\text{length}(\text{---}) = 33 \times 6 \text{ copies} = 198 \text{ nt}$$

# How can we perform inference from sequenced fragments?



In an unbiased experiment,  
sampling fragments depends on:

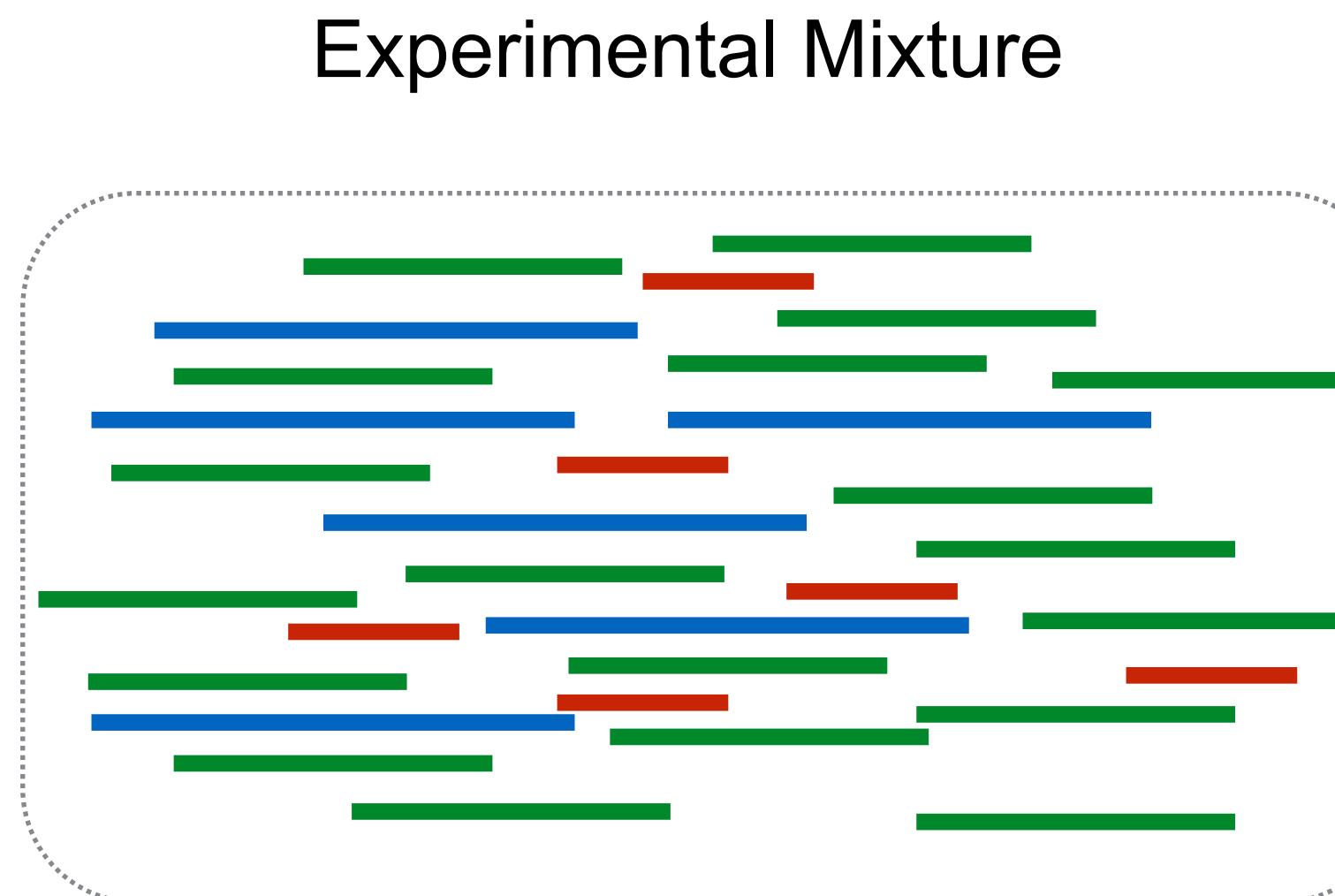
- # of copies of each txp type
- length of each txp type

$$\text{length}(\text{---}) = 100 \text{ nt} \times 6 \text{ copies} = 600 \text{ nt} \quad \sim 30\% \text{ blue}$$

$$\text{length}(\text{---}) = 66 \text{ nt} \times 19 \text{ copies} = 1254 \text{ nt} \quad \sim 60\% \text{ green}$$

$$\text{length}(\text{---}) = 33 \text{ nt} \times 6 \text{ copies} = 198 \text{ nt} \quad \sim 10\% \text{ red}$$

# How can we perform inference from sequenced fragments?



In an unbiased experiment,  
sampling fragments depends on:

- # of copies of each txp type
- length of each txp type

$$\text{length}(\text{--- blue ---}) = 100 \text{ nt} \times 6 \text{ copies} = 600 \text{ nt} \quad \sim 30\% \text{ blue}$$

$$\text{length}(\text{--- green ---}) = 66 \text{ nt} \times 19 \text{ copies} = 1254 \text{ nt} \quad \sim 60\% \text{ green}$$

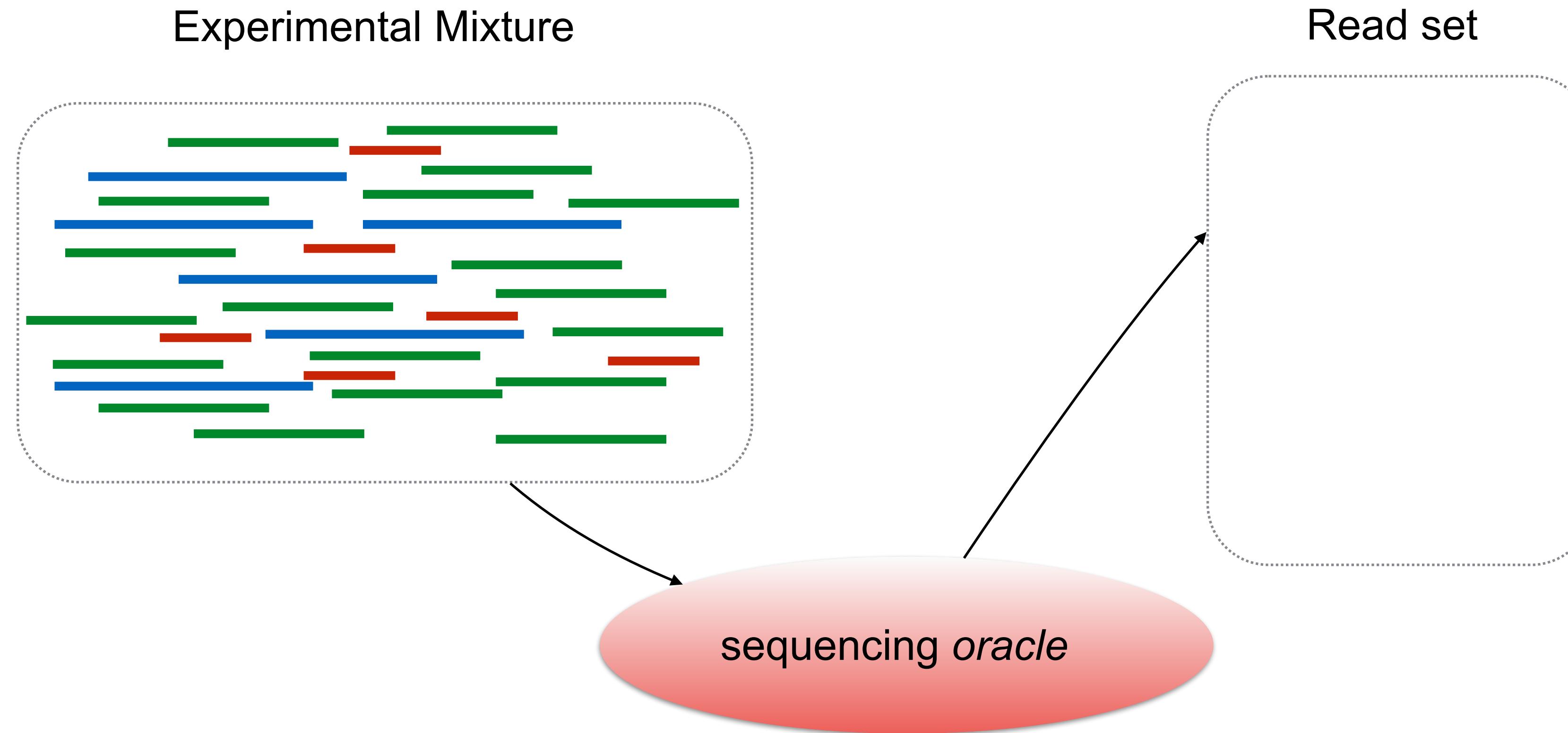
$$\text{length}(\text{--- red ---}) = 33 \text{ nt} \times 6 \text{ copies} = 198 \text{ nt} \quad \sim 10\% \text{ red}$$



We call these values  $\eta = [0.3, 0.6, 0.1]$  the nucleotide fractions,  
they become the primary quantity of interest

# How can we perform inference from sequenced fragments?

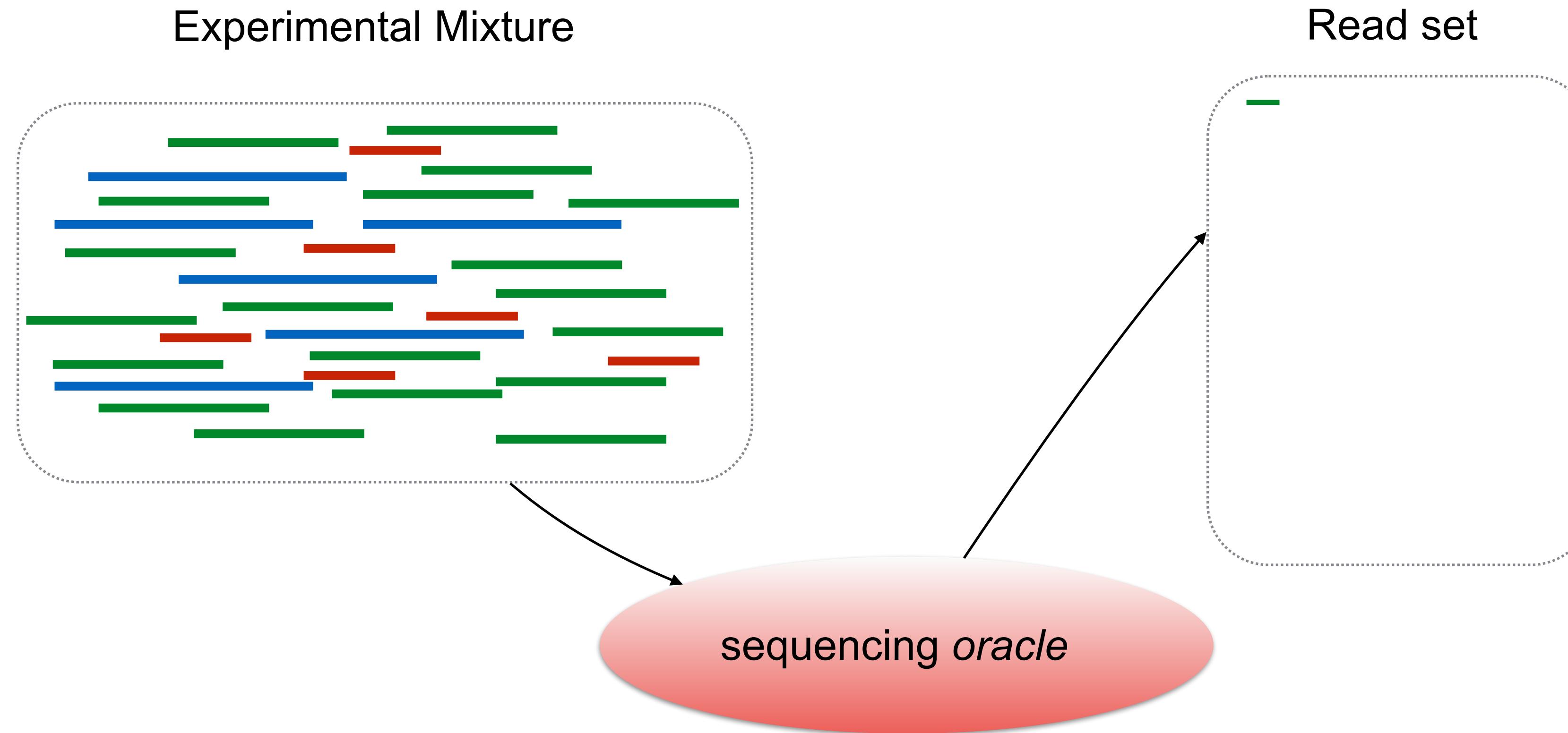
Think about the “ideal” RNA-seq experiment . . .



- (1) Pick transcript  $t \propto$  total available nucleotides = count \* length
- (2) Pick a position  $p$  on  $t$  “uniformly at random”

# How can we perform inference from sequenced fragments?

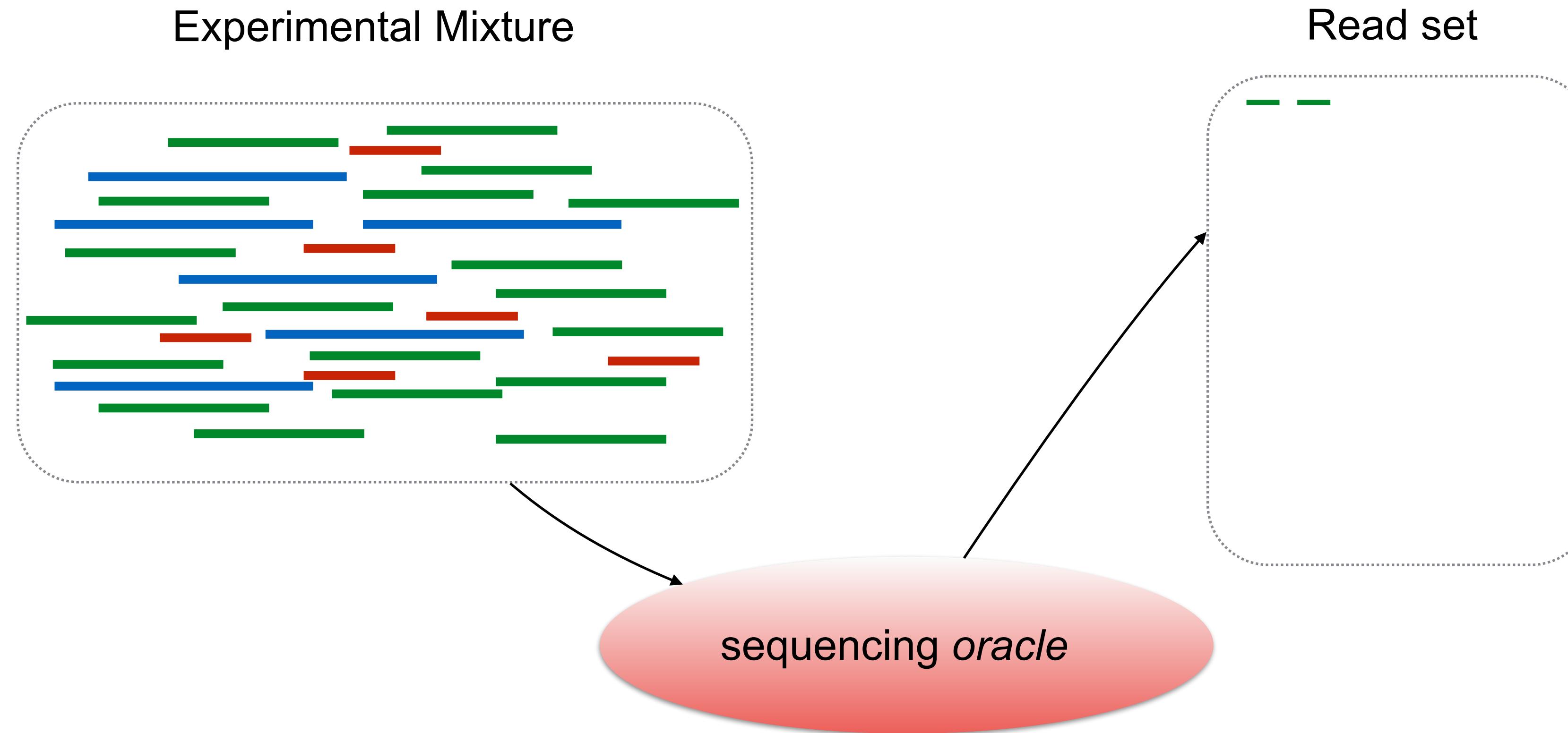
Think about the “ideal” RNA-seq experiment . . .



- (1) Pick transcript  $t \propto$  total available nucleotides = count \* length
- (2) Pick a position  $p$  on  $t$  “uniformly at random”

# How can we perform inference from sequenced fragments?

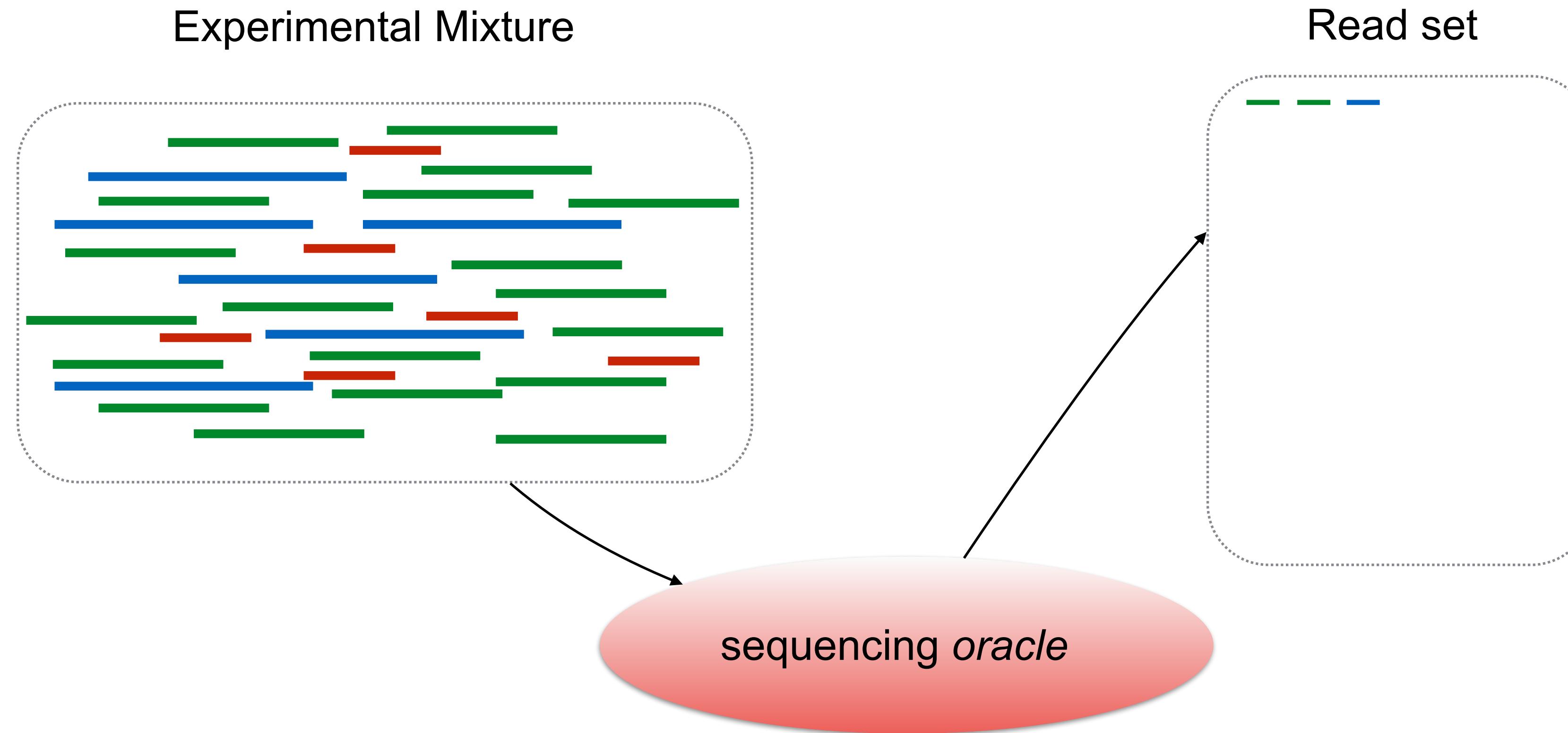
Think about the “ideal” RNA-seq experiment . . .



- (1) Pick transcript  $t \propto$  total available nucleotides = count \* length
- (2) Pick a position  $p$  on  $t$  “uniformly at random”

# How can we perform inference from sequenced fragments?

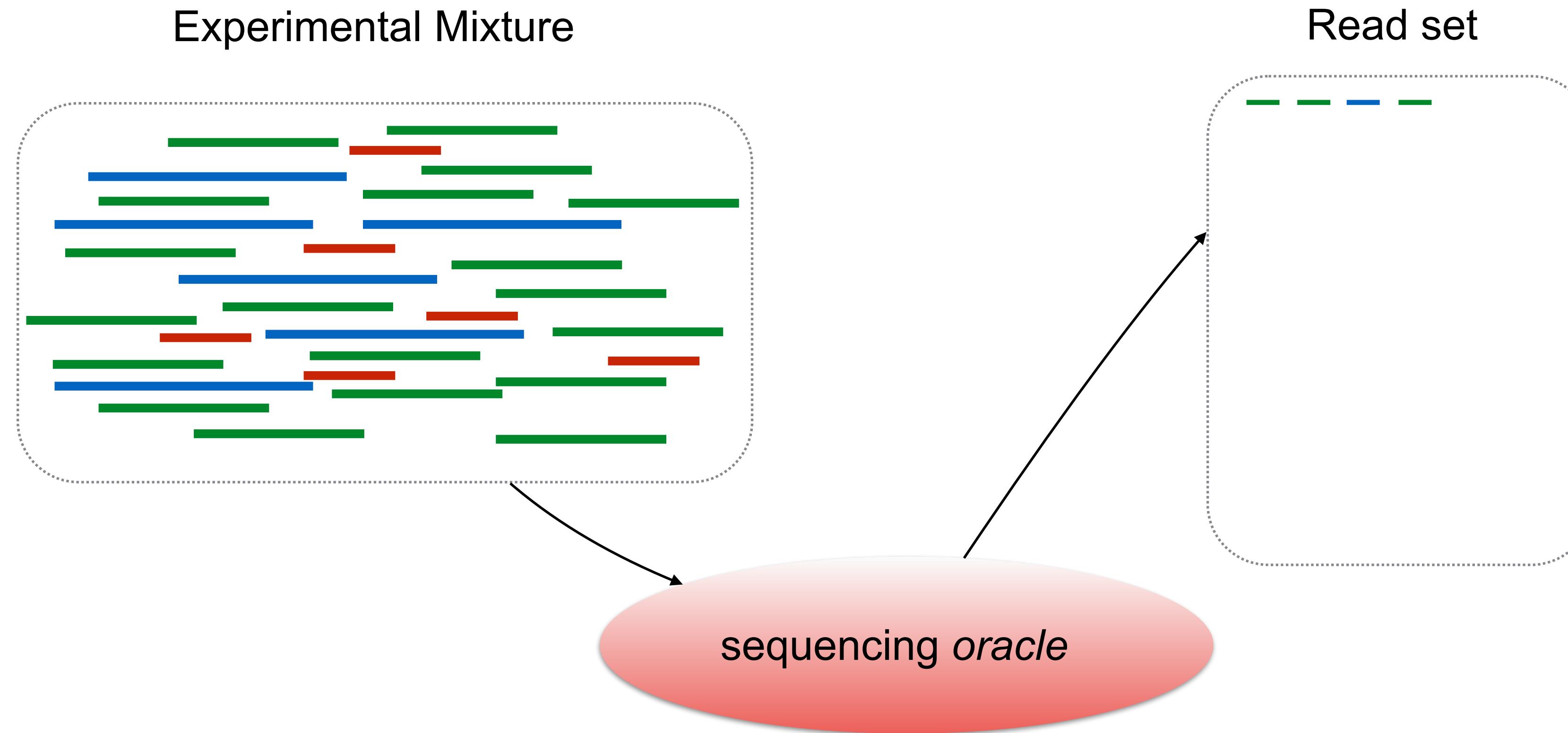
Think about the “ideal” RNA-seq experiment . . .



- (1) Pick transcript  $t \propto$  total available nucleotides = count \* length
- (2) Pick a position  $p$  on  $t$  “uniformly at random”

# How can we perform inference from sequenced fragments?

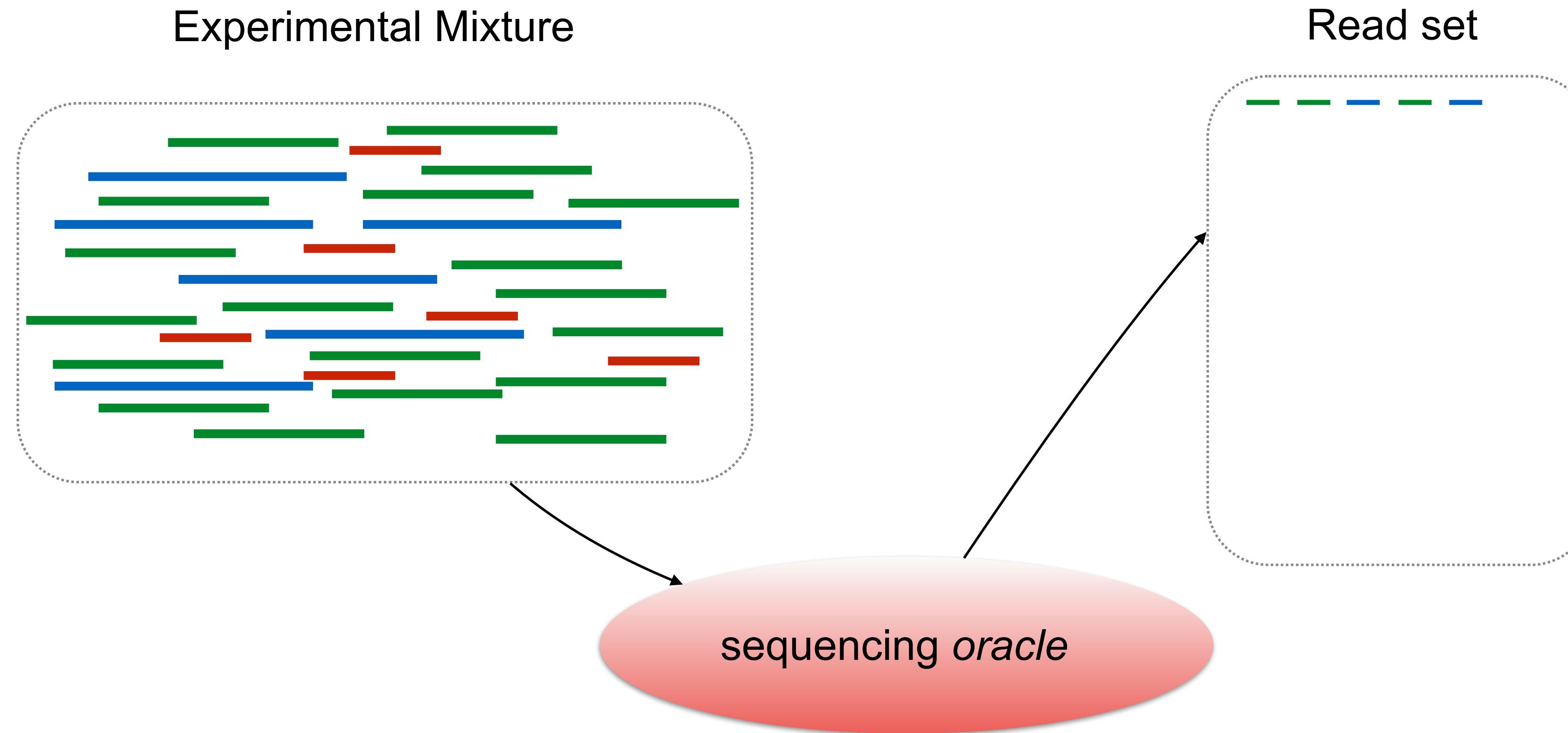
Think about the “ideal” RNA-seq experiment . . .



- (1) Pick transcript  $t \propto$  total available nucleotides = count \* length
- (2) Pick a position  $p$  on  $t$  “uniformly at random”

# How can we perform inference from sequenced fragments?

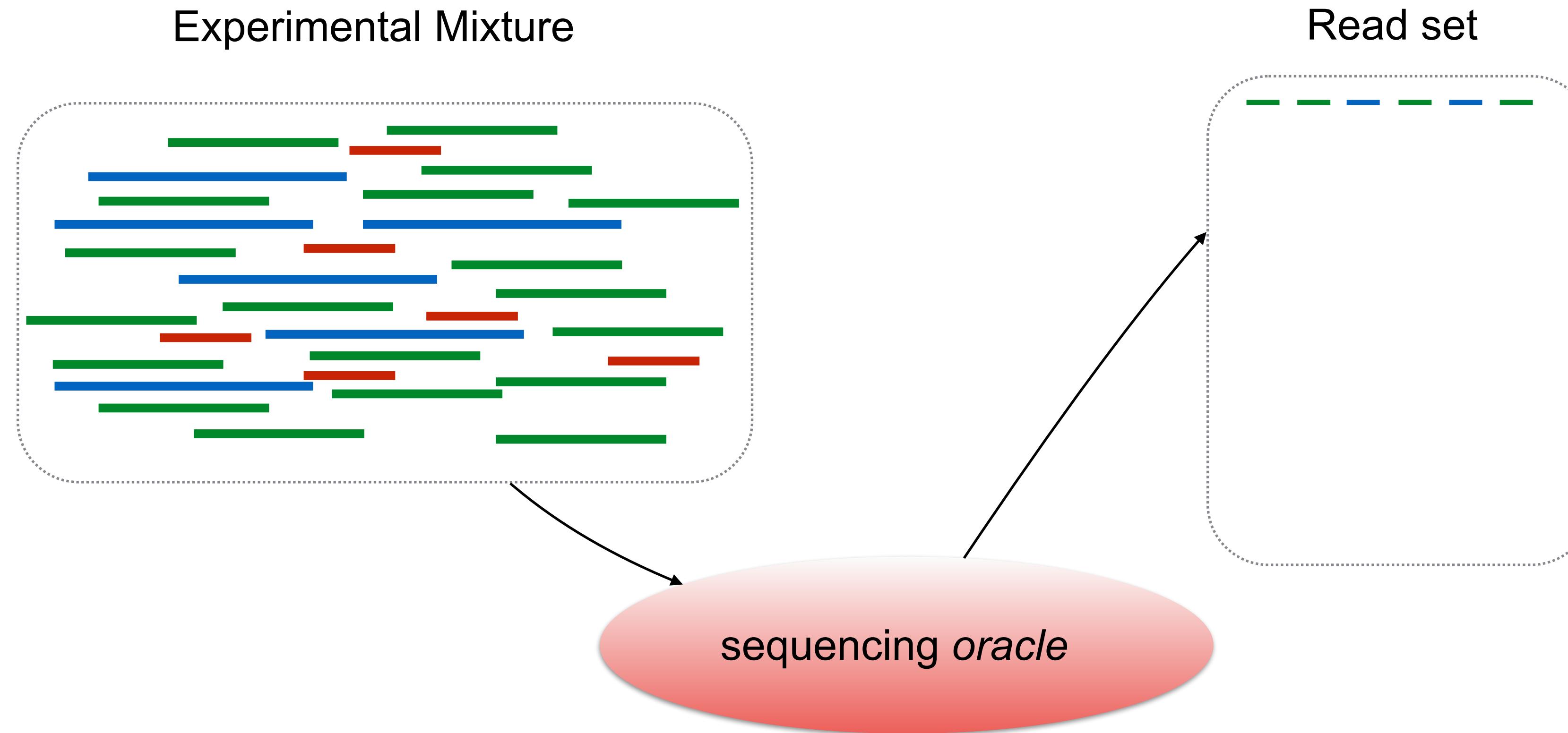
Think about the “ideal” RNA-seq experiment . . .



- (1) Pick transcript  $t \propto$  total available nucleotides = count \* length
- (2) Pick a position  $p$  on  $t$  “uniformly at random”

# How can we perform inference from sequenced fragments?

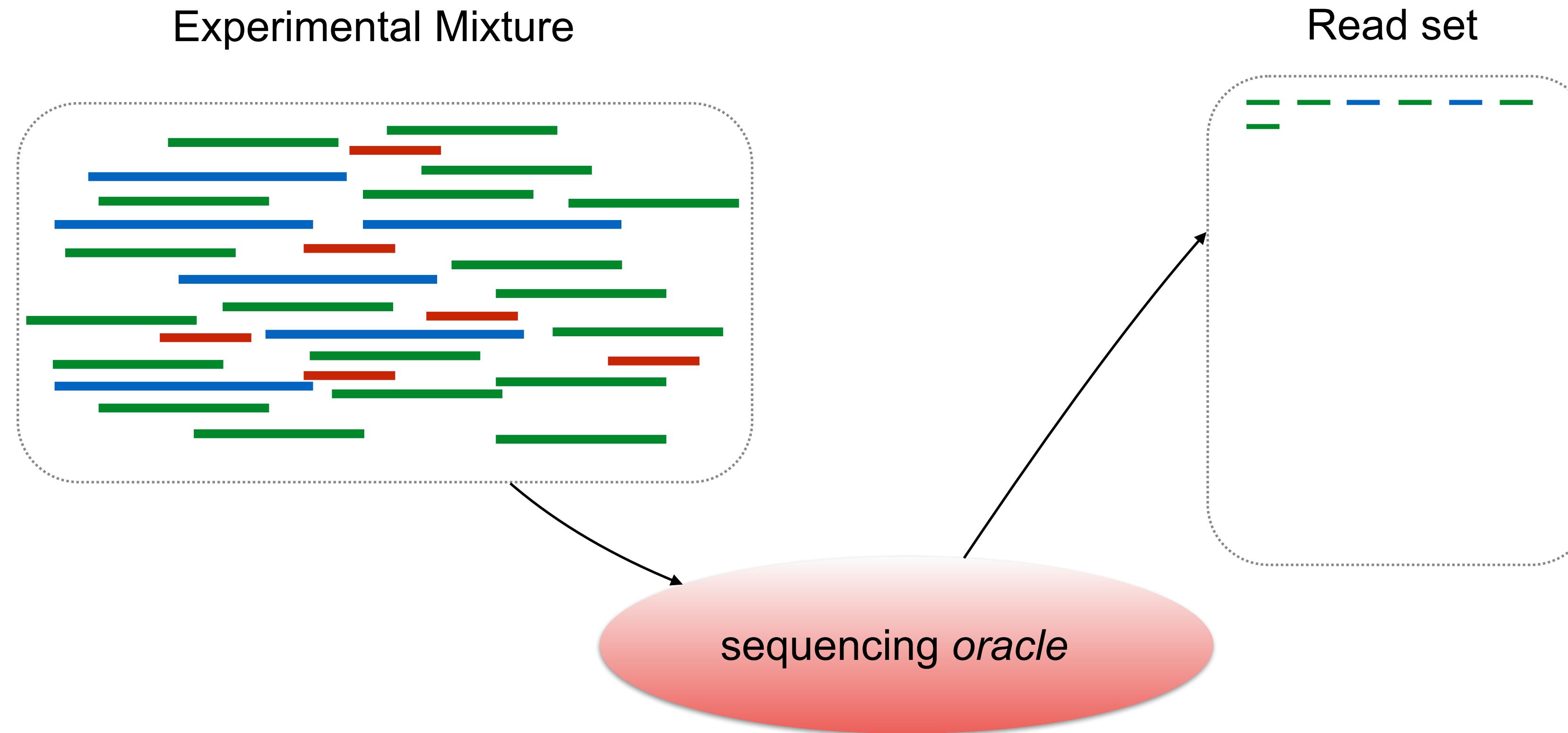
Think about the “ideal” RNA-seq experiment . . .



- (1) Pick transcript  $t \propto$  total available nucleotides = count \* length
- (2) Pick a position  $p$  on  $t$  “uniformly at random”

# How can we perform inference from sequenced fragments?

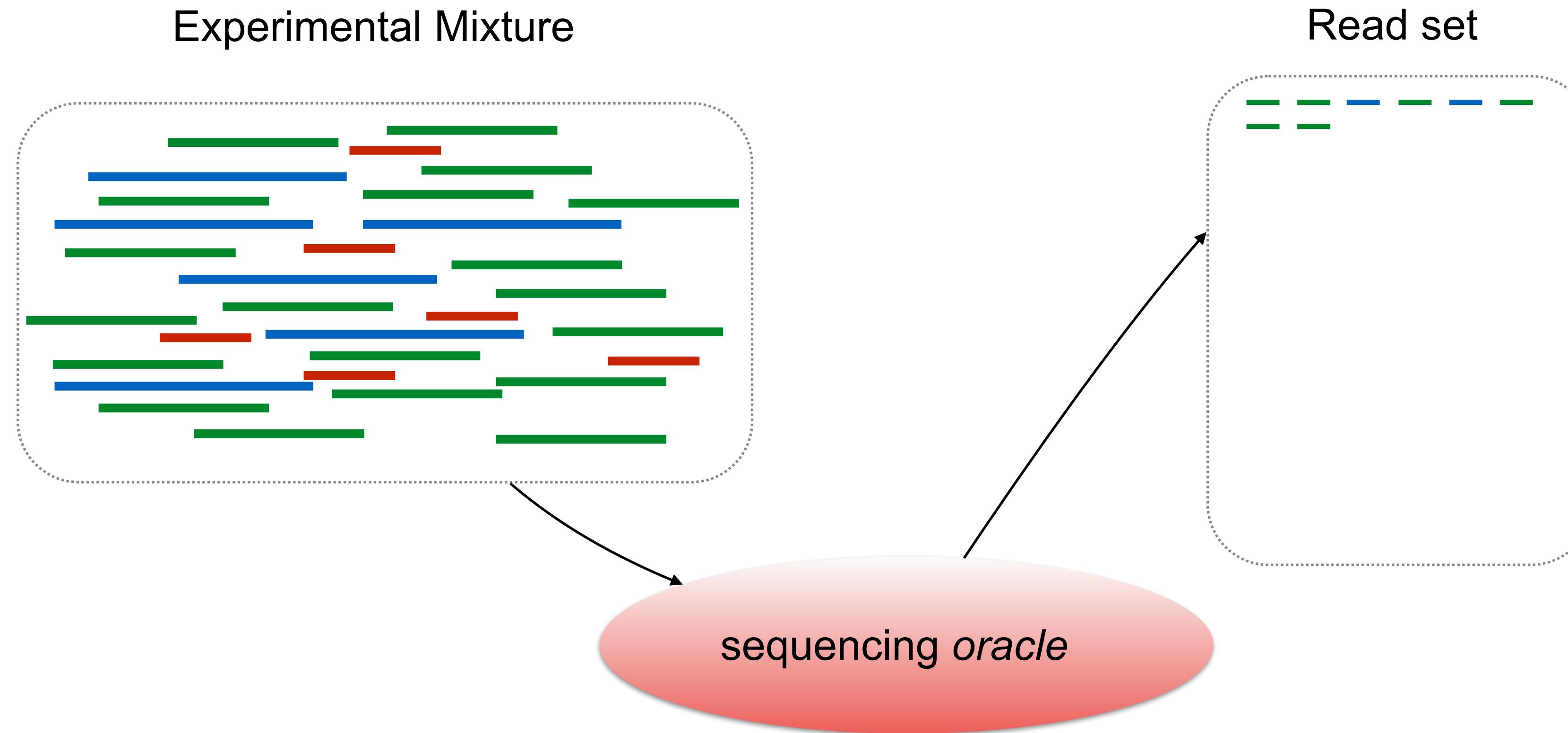
Think about the “ideal” RNA-seq experiment . . .



- (1) Pick transcript  $t \propto$  total available nucleotides = count \* length
- (2) Pick a position  $p$  on  $t$  “uniformly at random”

# How can we perform inference from sequenced fragments?

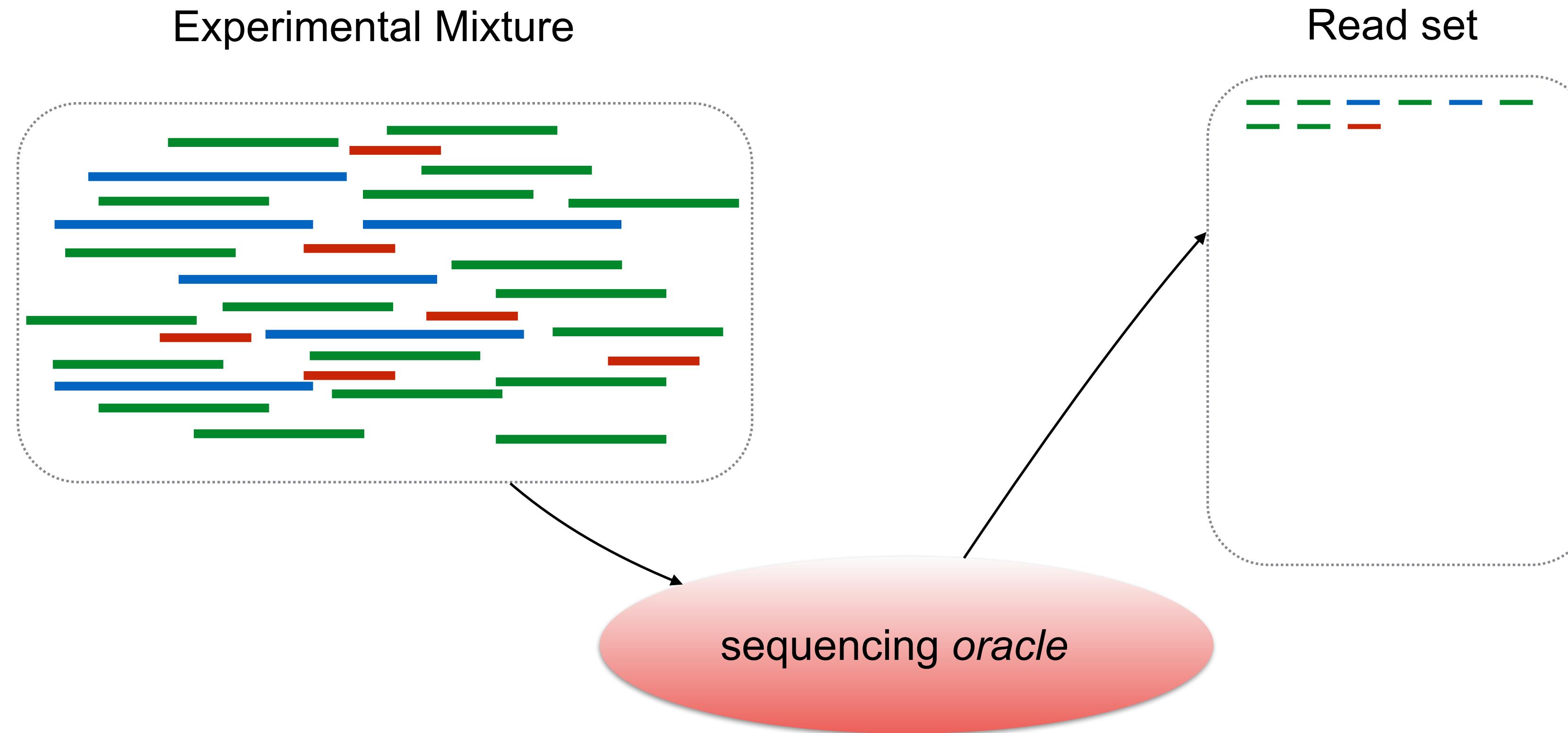
Think about the “ideal” RNA-seq experiment . . .



- (1) Pick transcript  $t \propto$  total available nucleotides = count \* length
- (2) Pick a position  $p$  on  $t$  “uniformly at random”

# How can we perform inference from sequenced fragments?

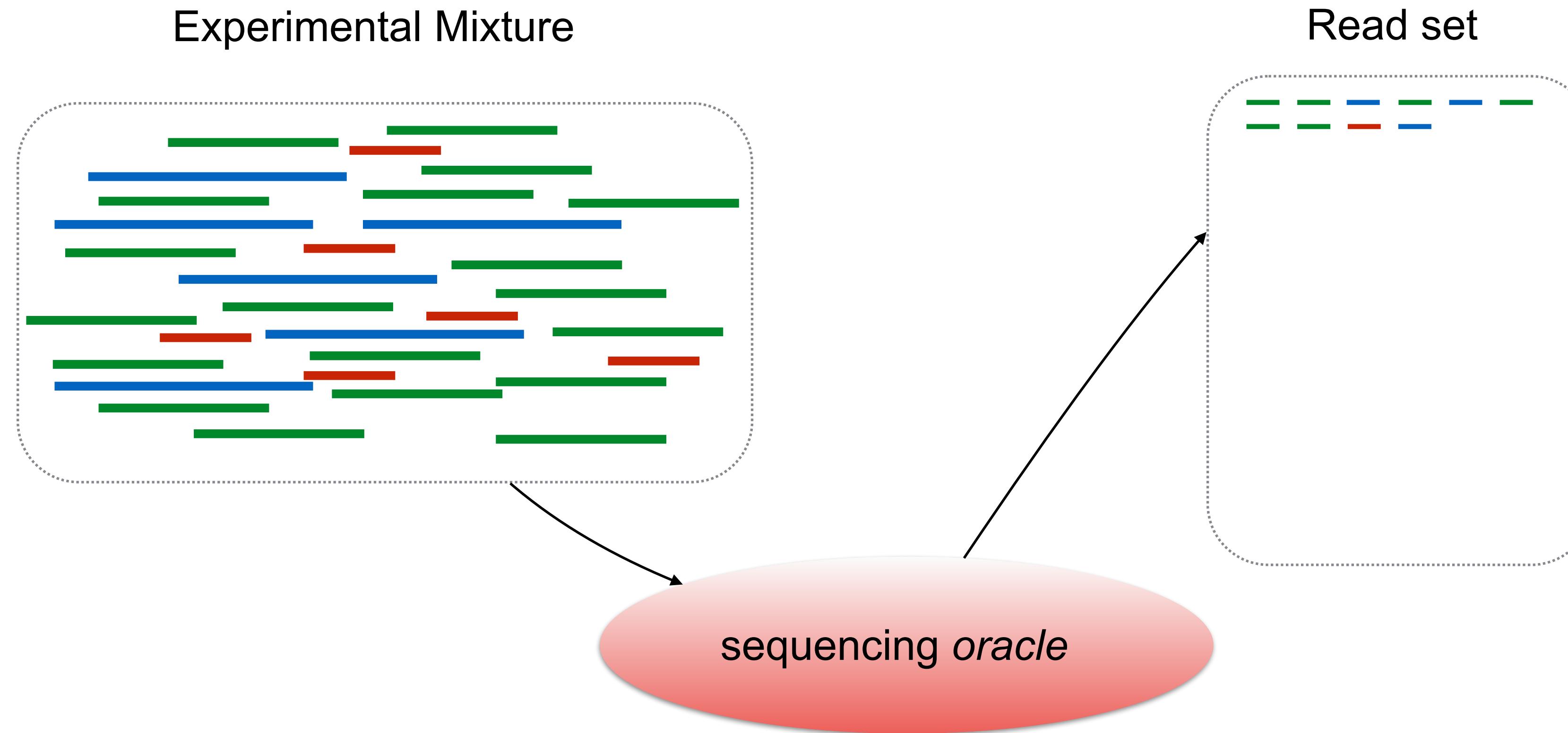
Think about the “ideal” RNA-seq experiment . . .



- (1) Pick transcript  $t \propto$  total available nucleotides = count \* length
- (2) Pick a position  $p$  on  $t$  “uniformly at random”

# How can we perform inference from sequenced fragments?

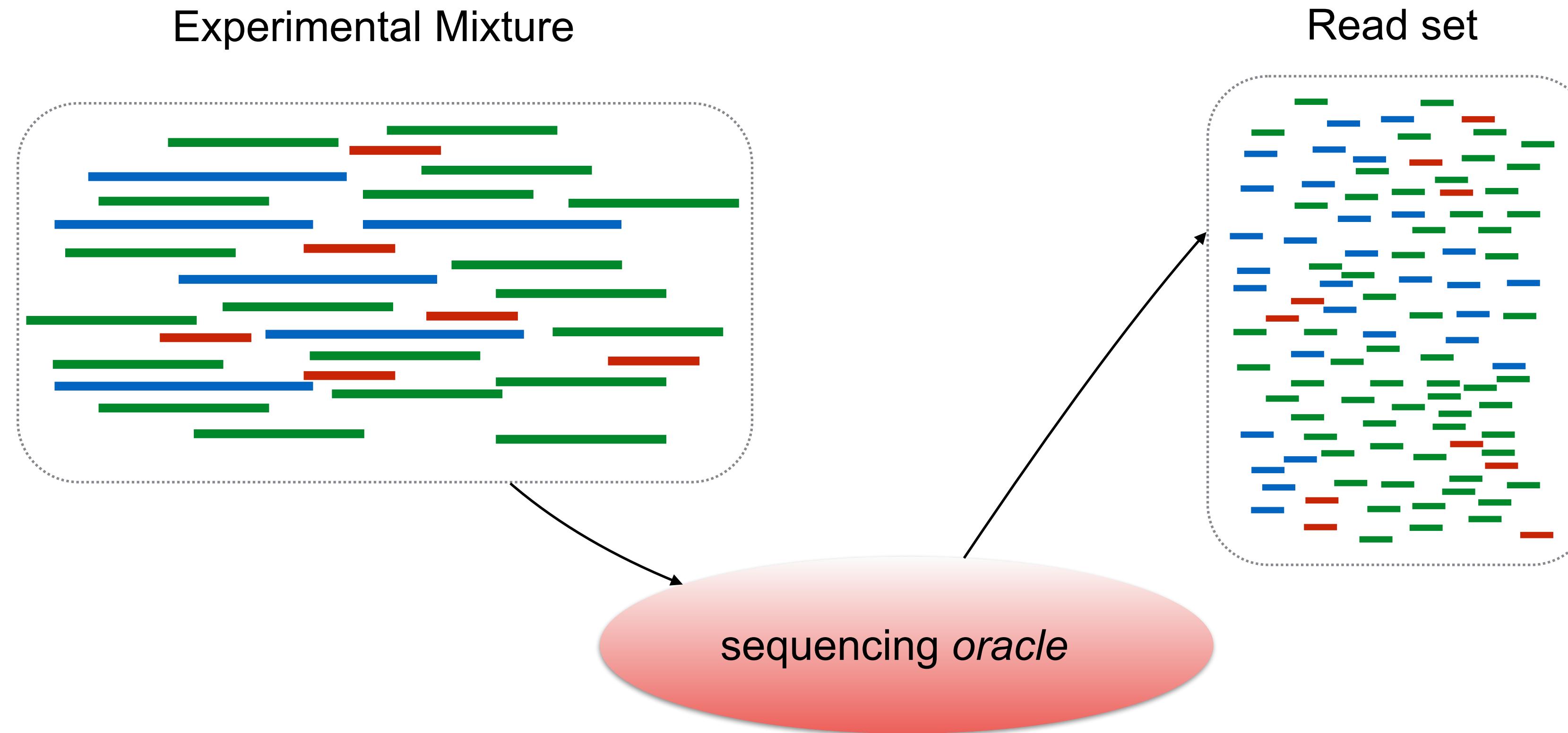
Think about the “ideal” RNA-seq experiment . . .



- (1) Pick transcript  $t \propto$  total available nucleotides = count \* length
- (2) Pick a position  $p$  on  $t$  “uniformly at random”

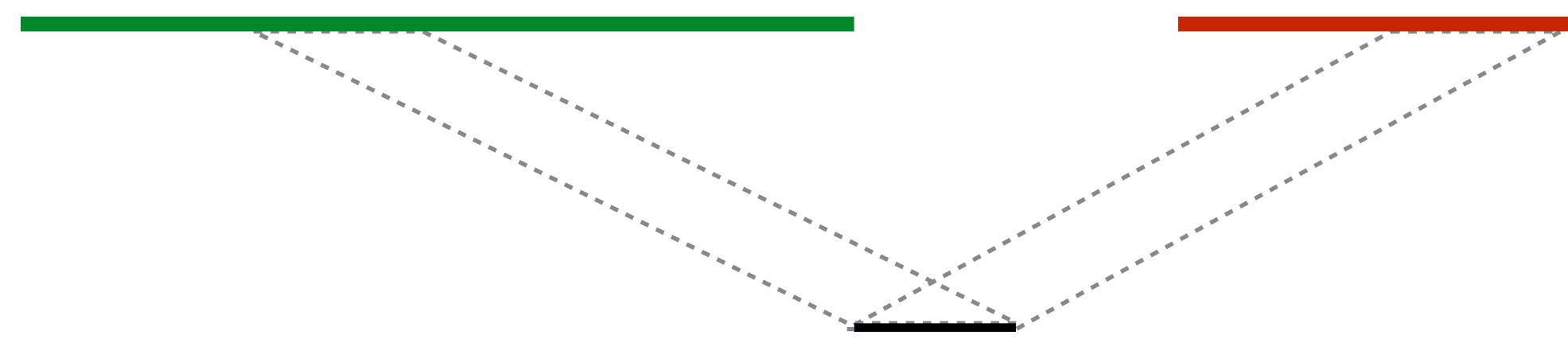
# How can we perform inference from sequenced fragments?

Think about the “ideal” RNA-seq experiment . . .



- (1) Pick transcript  $t \propto$  total available nucleotides = count \* length
- (2) Pick a position  $p$  on  $t$  “uniformly at random”

# Resolving a single multi-mapping read



Say we *knew* the  $\eta$ , and observed a *single* read that mapped ambiguously, as shown above.

What is the probability that it truly originated from **G** or **R**?

$$\Pr \{r \text{ from } G\} = \frac{\frac{\eta_G}{\text{length}(G)}}{\frac{\eta_G}{\text{length}(G)} + \frac{\eta_R}{\text{length}(R)}} = \frac{\frac{0.6}{66}}{\frac{0.6}{66} + \frac{0.1}{33}} = 0.75$$

normalization factor

$$\Pr \{r \text{ from } R\} = \frac{\frac{\eta_R}{\text{length}(R)}}{\frac{\eta_G}{\text{length}(G)} + \frac{\eta_R}{\text{length}(R)}} = \frac{\frac{0.1}{33}}{\frac{0.6}{66} + \frac{0.1}{33}} = 0.25$$

$$\text{length}( \text{blue line} ) = 100 \times 6 \text{ copies} = 600 \text{ nt} \sim 30\% \text{ blue}$$

$$\text{length}( \text{green line} ) = 66 \times 19 \text{ copies} = 1254 \text{ nt} \sim 60\% \text{ green}$$

$$\text{length}( \text{red line} ) = 33 \times 6 \text{ copies} = 198 \text{ nt} \sim 10\% \text{ red}$$

# Units for Relative Abundance

TPM (Transcripts Per Million)

$$\text{TPM}_i = \rho_i \times 10^6 \text{ where } 0 \leq \rho_i \leq 1 \text{ and } \sum_i \rho_i = 1$$

$$\rho_i = \frac{\frac{X_i}{\ell_i}}{\sum_j \frac{X_j}{\ell_j}}$$

Reads coming from  
transcript i

# Units for Relative Abundance

TPM (Transcripts Per Million)

$$\text{TPM}_i = \rho_i \times 10^6 \text{ where } 0 \leq \rho_i \leq 1 \text{ and } \sum_i \rho_i = 1$$

$$\rho_i = \frac{\frac{X_i}{\ell_i}}{\sum_j \frac{X_j}{\ell_j}}$$

Reads coming from  
transcript i

Length of transcript i

# Units for Relative Abundance

TPM (Transcripts Per Million)

$$\text{TPM}_i = \rho_i \times 10^6 \text{ where } 0 \leq \rho_i \leq 1 \text{ and } \sum_i \rho_i = 1$$

abundance of i  
as fraction of all  
measured transcripts

$$\rho_i = \frac{\frac{X_i}{\ell_i}}{\sum_j \frac{X_j}{\ell_j}}$$

Reads coming from  
transcript i

Length of transcript i

# Units for Relative Abundance

TPM (Transcripts Per Million)

$$\text{TPM}_i = \rho_i \times 10^6 \text{ where } 0 \leq \rho_i \leq 1 \text{ and } \sum_i \rho_i = 1$$

abundance of i  
as fraction of all  
measured transcripts

$$\rho_i = \frac{\frac{X_i}{\ell_i}}{\sum_j \frac{X_j}{\ell_j}}$$

Reads coming from  
transcript i

Length of transcript i

# Aside: Maximum Likelihood Est. and the EM Algorithm

**The following slides on MLE & EM are taken from the UW CSE 312 Web\***

# Parameter Estimation

Assuming sample  $x_1, x_2, \dots, x_n$  is from a parametric distribution  $f(x|\theta)$ , estimate  $\theta$ .

E.g.: Given sample HHTTTTHTHTHTHH  
of (possibly biased) coin flips, estimate

$\theta$  = probability of Heads

$f(x|\theta)$  is the Bernoulli probability mass function with parameter  $\theta$

# Likelihood

$P(x | \theta)$ : Probability of event  $x$  given model  $\theta$

Viewed as a function of  $x$  (fixed  $\theta$ ), it's a *probability*

E.g.,  $\sum_x P(x | \theta) = 1$

Viewed as a function of  $\theta$  (fixed  $x$ ), it's a *likelihood*

E.g.,  $\sum_\theta P(x | \theta)$  can be anything; *relative* values of interest.

E.g., if  $\theta$  = prob of heads in a sequence of coin flips then

$P(HHTHH | .6) > P(HHTHH | .5)$ ,

I.e., event HHTHH is *more likely* when  $\theta = .6$  than  $\theta = .5$

And what  $\theta$  make HHTHH *most likely*?

# Likelihood

$P(x | \theta)$ : Probability of event  $x$  given model  $\theta$

Viewed as a function of  $x$  (fixed  $\theta$ ), it's a *probability*

E.g.,  $\sum_x P(x | \theta) = 1$

Viewed as a function of  $\theta$  (fixed  $x$ ), it's a *likelihood*

E.g.,  $\sum_\theta P(x | \theta)$  can be anything; *relative* values of interest.

E.g., if  $\theta$  = prob of heads in a sequence of coin flips then

$P(HHTHH | .6) > P(HHTHH | .5)$ ,

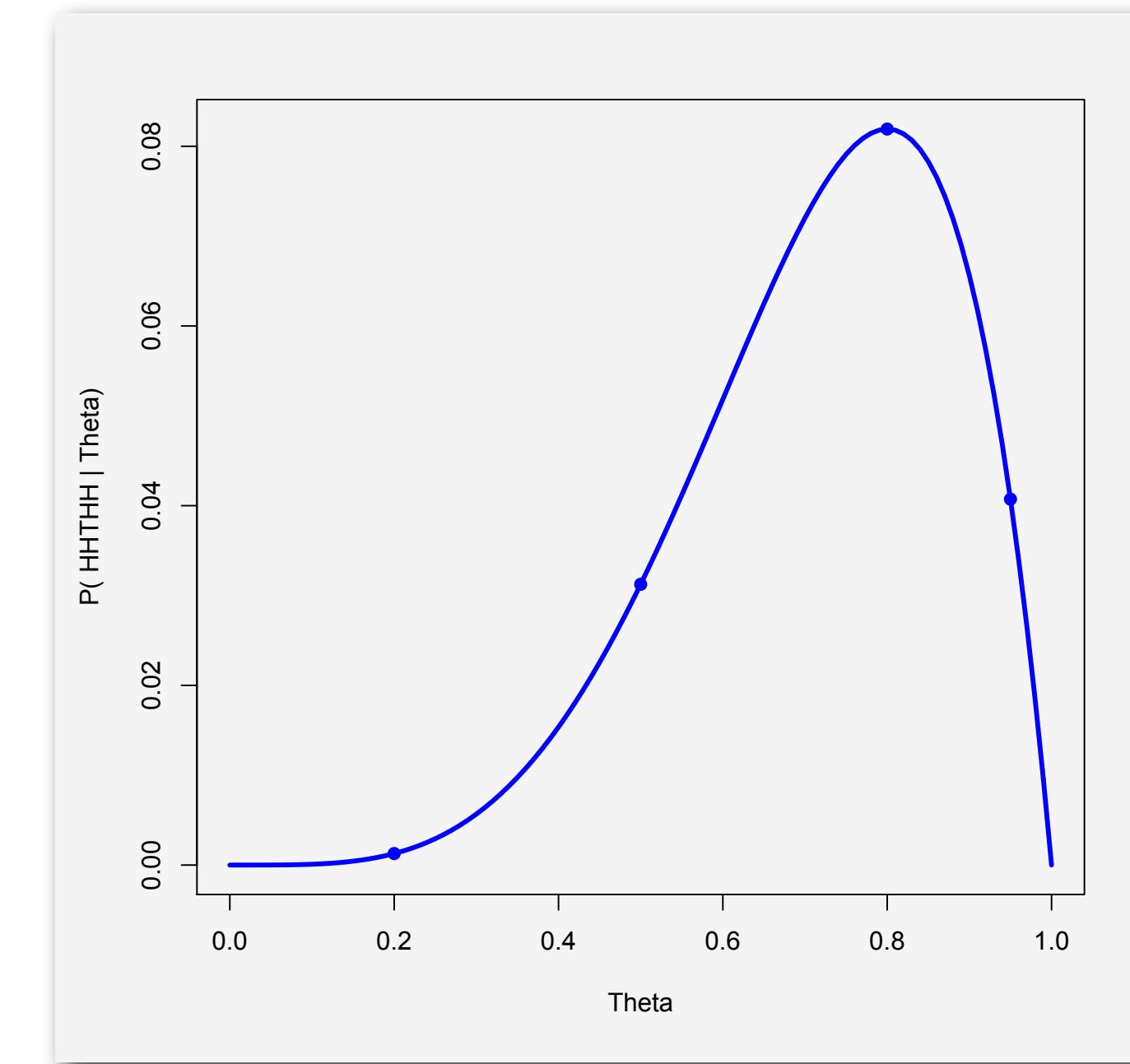
I.e., event HHTHH is *more likely* when  $\theta = .6$  than  $\theta = .5$

And what  $\theta$  make HHTHH *most likely*?

# Likelihood Function

Probability of HHTHH,  
given  $P(H) = \theta$ :

$\theta$	$\theta^4(1-\theta)$
0.2	0.0013
0.5	0.0313
0.8	0.0819
0.95	0.0407



# Maximum Likelihood Parameter Estimation

One (of many) approaches to param. est.

Likelihood of (indp) observations  $x_1, x_2, \dots, x_n$

$$L(x_1, x_2, \dots, x_n \mid \theta) = \prod_{i=1}^n f(x_i \mid \theta)$$

As a function of  $\theta$ , what  $\theta$  maximizes the likelihood of the data actually observed

Typical approach:  $\frac{\partial}{\partial \theta} L(\vec{x} \mid \theta) = 0$  or  $\frac{\partial}{\partial \theta} \log L(\vec{x} \mid \theta) = 0$

# Example I

$n$  coin flips,  $x_1, x_2, \dots, x_n$ ;  $n_0$  tails,  $n_1$  heads,  $n_0 + n_1 = n$ ;

$\theta$  = probability of heads

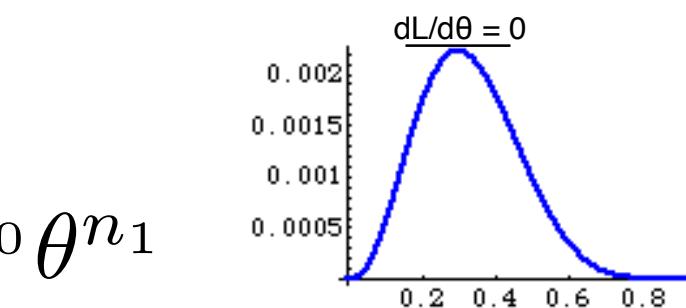
$$L(x_1, x_2, \dots, x_n \mid \theta) = (1 - \theta)^{n_0} \theta^{n_1}$$

$$\log L(x_1, x_2, \dots, x_n \mid \theta) = n_0 \log(1 - \theta) + n_1 \log \theta$$

$$\frac{\partial}{\partial \theta} \log L(x_1, x_2, \dots, x_n \mid \theta) = \frac{-n_0}{1-\theta} + \frac{n_1}{\theta}$$

Setting to zero and solving:

$$\hat{\theta} = \frac{n_1}{n}$$



Observed fraction of successes in sample is MLE of success probability in population

(Also verify it's max, not min, & not better on boundary)

# Bias

A desirable property: An estimator  $Y$  of a parameter  $\theta$  is an *unbiased* estimator if

$$E[Y] = \theta$$

For coin ex. above, MLE is unbiased:

$$Y = \text{fraction of heads} = (\sum_{1 \leq i \leq n} X_i) / n,$$

( $X_i$  = indicator for heads in  $i^{\text{th}}$  trial) so

$$E[Y] = (\sum_{1 \leq i \leq n} E[X_i]) / n = n \theta / n = \theta$$

## Aside: are all unbiased estimators equally good?

- No!
- E.g., “Ignore all but 1st flip; if it was H, let  $Y' = 1$ ; else  $Y' = 0$ ”
- Exercise: show this is unbiased
- Exercise: if observed data has at least one H and at least one T, what is the likelihood of the data given the model with  $\theta = Y'$  ?

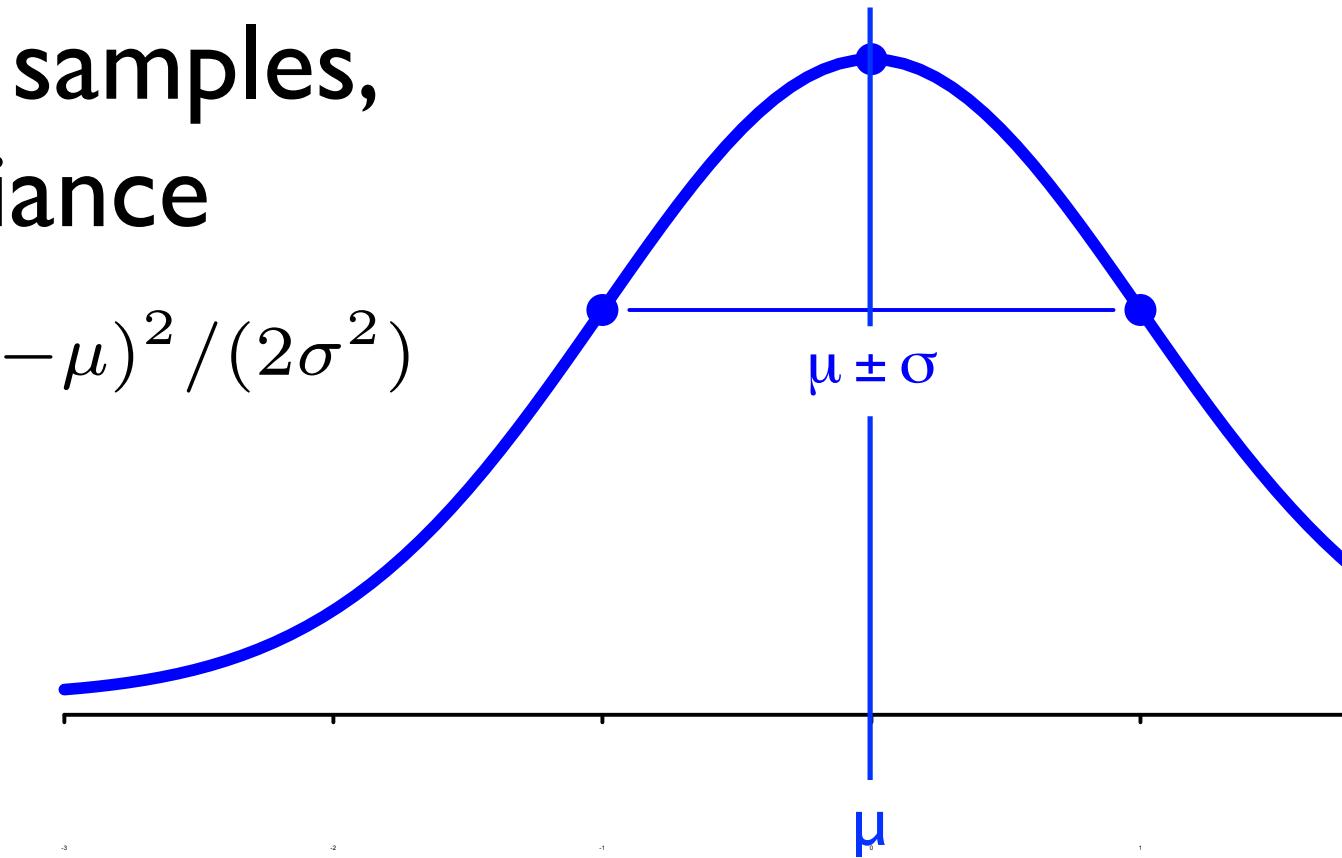
# Parameter Estimation

Assuming sample  $x_1, x_2, \dots, x_n$  is from a parametric distribution  $f(x|\theta)$ , estimate  $\theta$ .

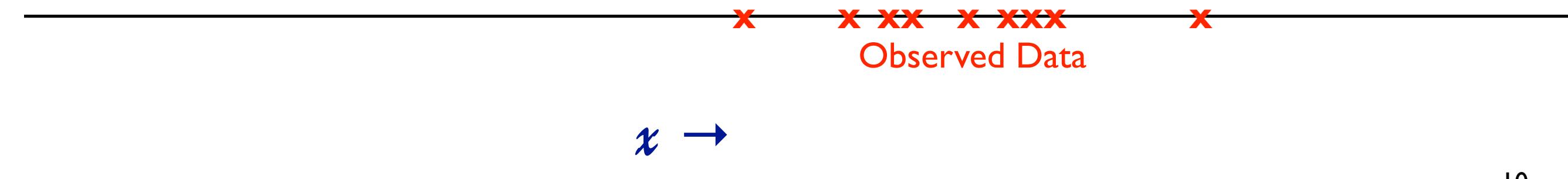
E.g.: Given  $n$  normal samples, estimate mean & variance

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)}$$

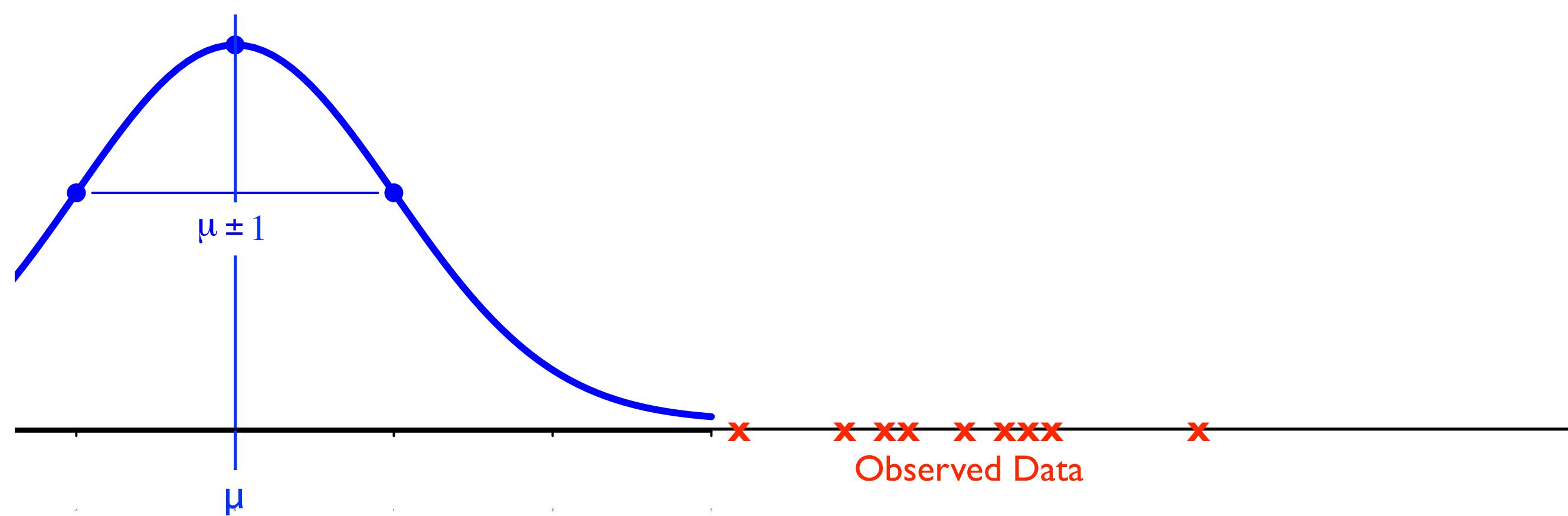
$$\theta = (\mu, \sigma^2)$$



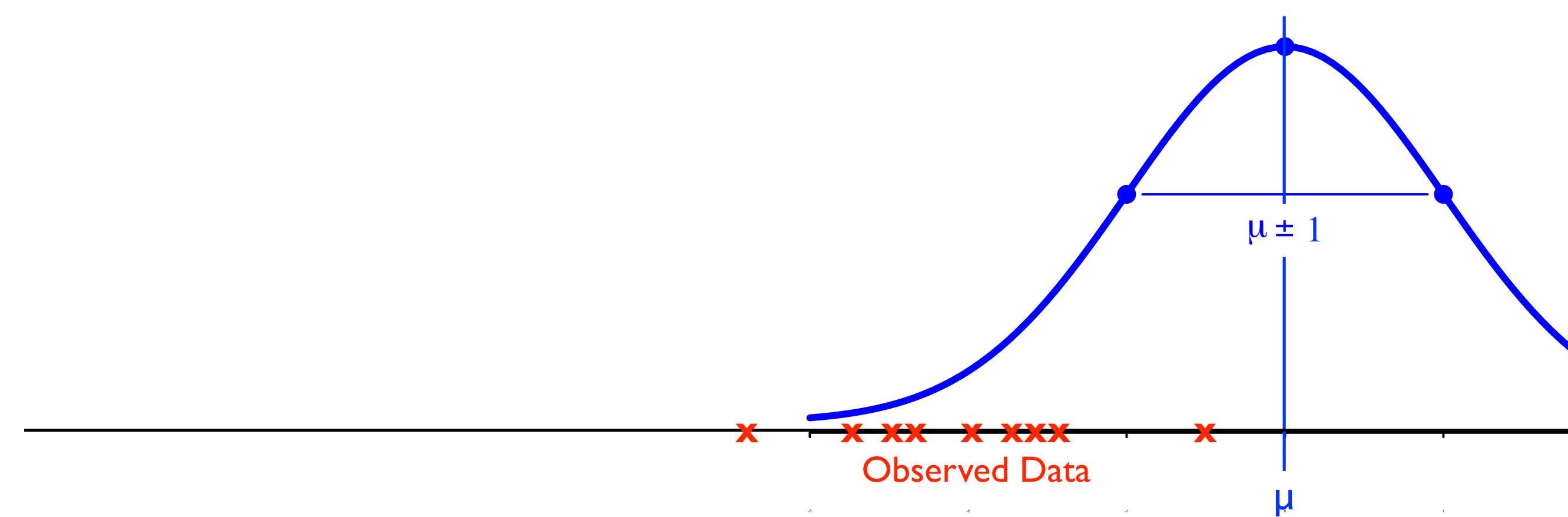
Ex2: I got data; a little birdie tells me  
it's normal, and promises  $\sigma^2 = 1$



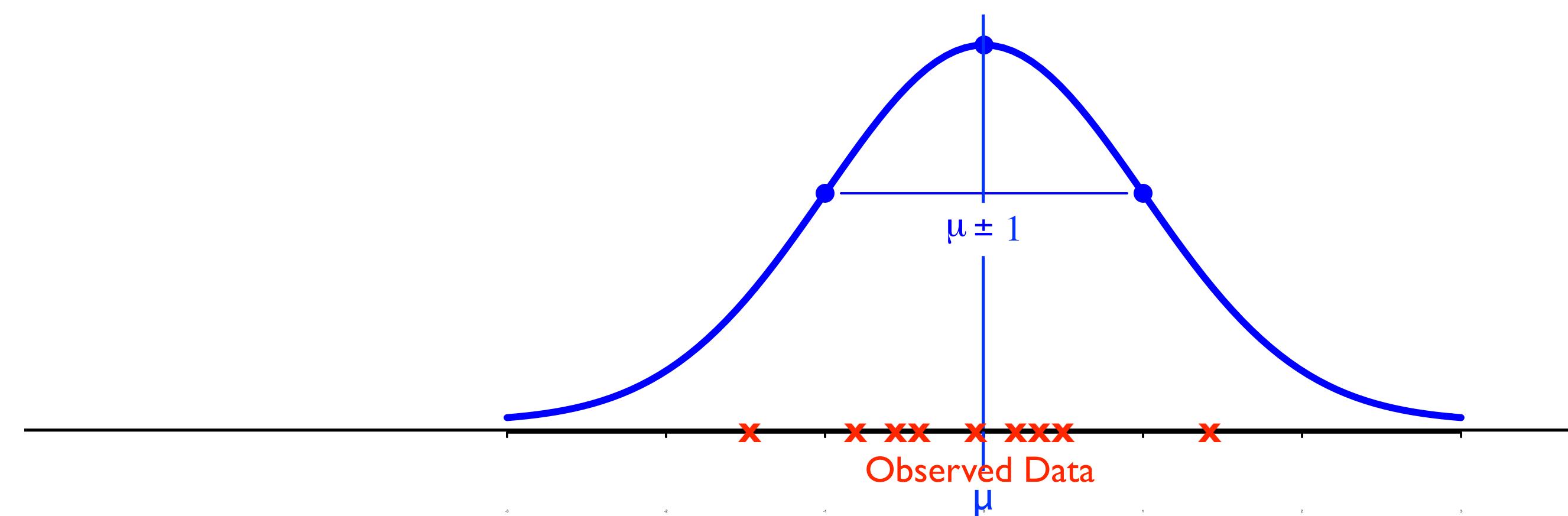
Which is more likely: (a) this?



Which is more likely: (b) or this?

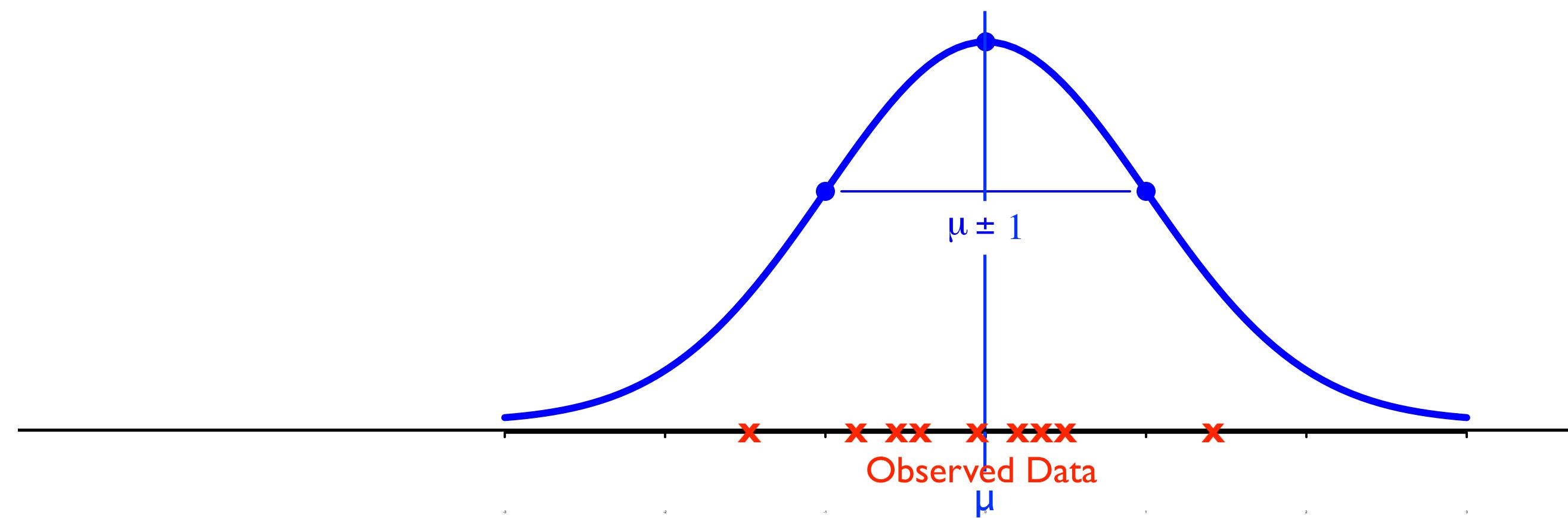


Which is more likely: (c) or *this*?



# Which is more likely: (c) or this?

Looks good by eye, but how do I optimize my estimate of  $\mu$  ?



**Ex. 2:**  $x_i \sim N(\mu, \sigma^2)$ ,  $\sigma^2 = 1$ ,  $\mu$  unknown

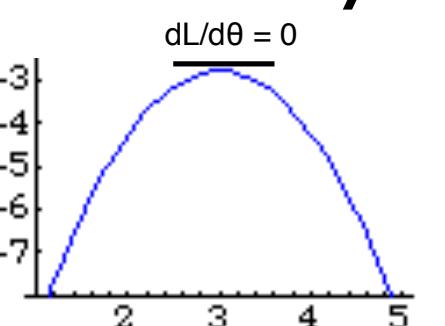
$$L(x_1, x_2, \dots, x_n | \theta) = \prod_{1 \leq i \leq n} \frac{1}{\sqrt{2\pi}} e^{-(x_i - \theta)^2/2}$$

$$\ln L(x_1, x_2, \dots, x_n | \theta) = \sum_{1 \leq i \leq n} -\frac{1}{2} \ln 2\pi - \frac{(x_i - \theta)^2}{2}$$

$$\frac{d}{d\theta} \ln L(x_1, x_2, \dots, x_n | \theta) = \sum_{1 \leq i \leq n} (x_i - \theta)$$

$$= \left( \sum_{1 \leq i \leq n} x_i \right) - n\theta = 0$$

And verify it's max,  
not min & not better  
on boundary



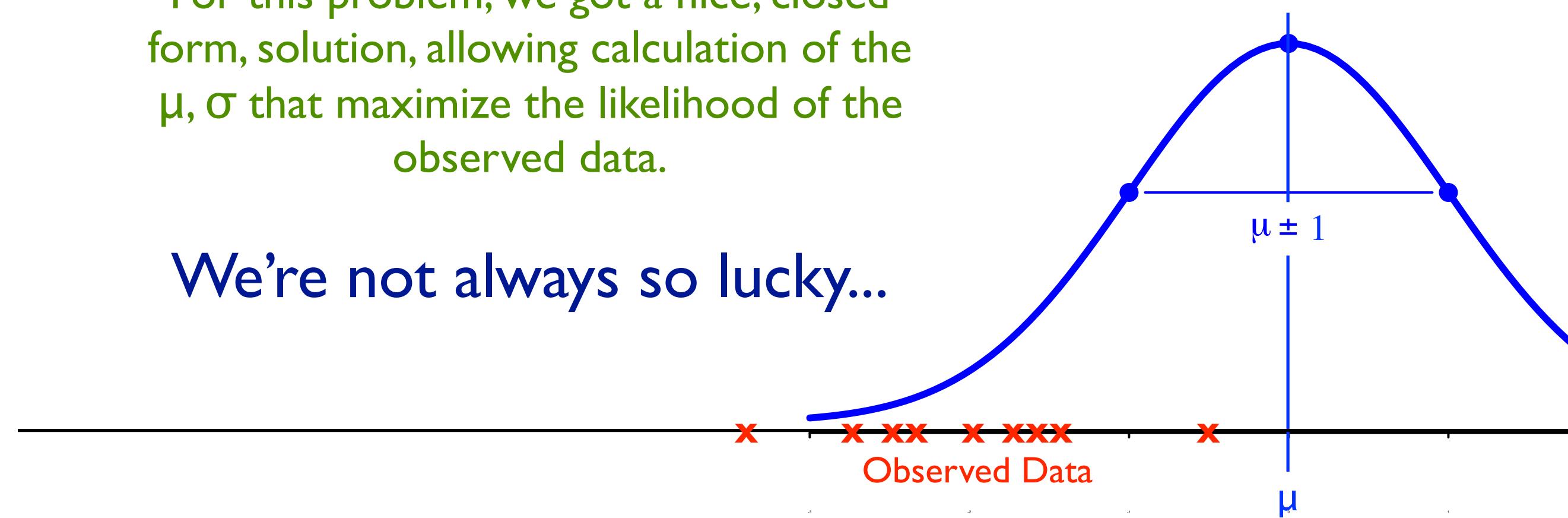
$$\hat{\theta} = \left( \sum_{1 \leq i \leq n} x_i \right) / n = \bar{x}$$

Sample mean is MLE of  
population mean

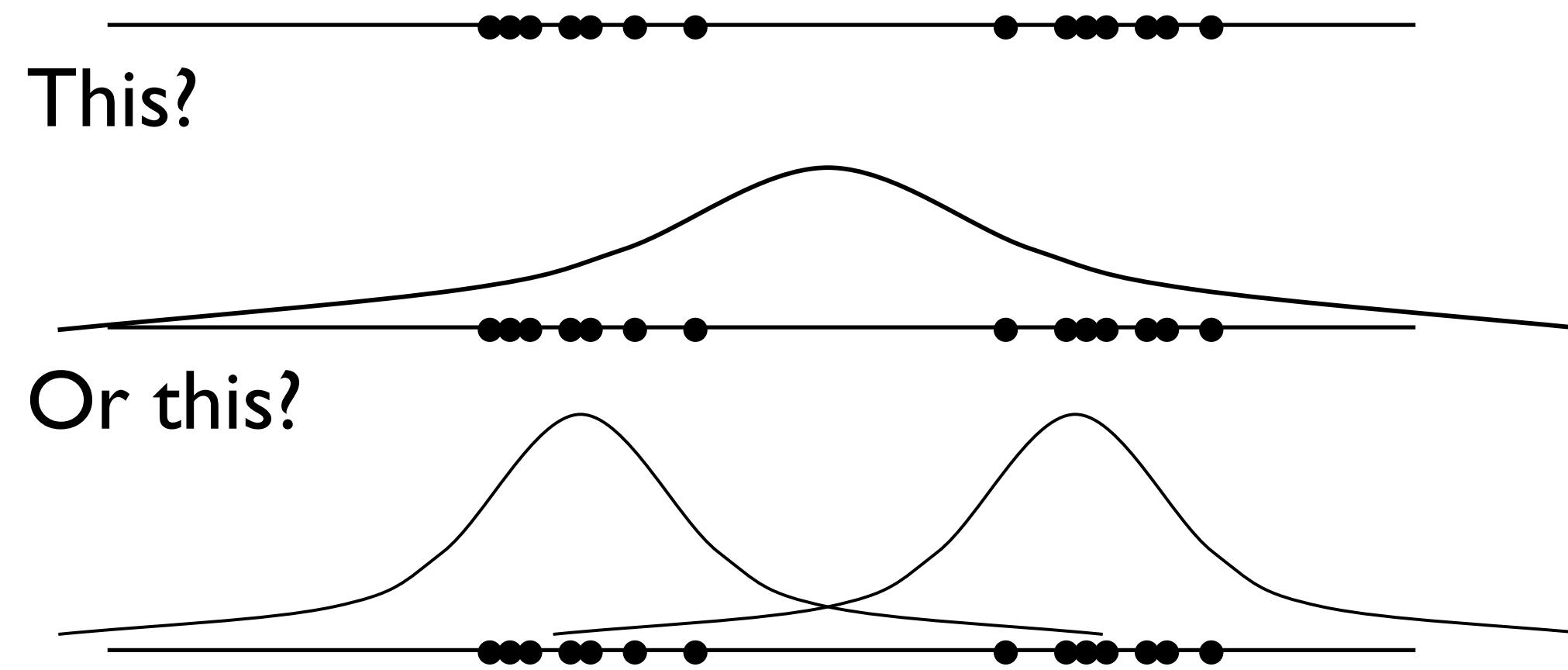
# Last lecture: How to estimate $\mu$ given data

For this problem, we got a nice, closed form, solution, allowing calculation of the  $\mu, \sigma$  that maximize the likelihood of the observed data.

We're not always so lucky...

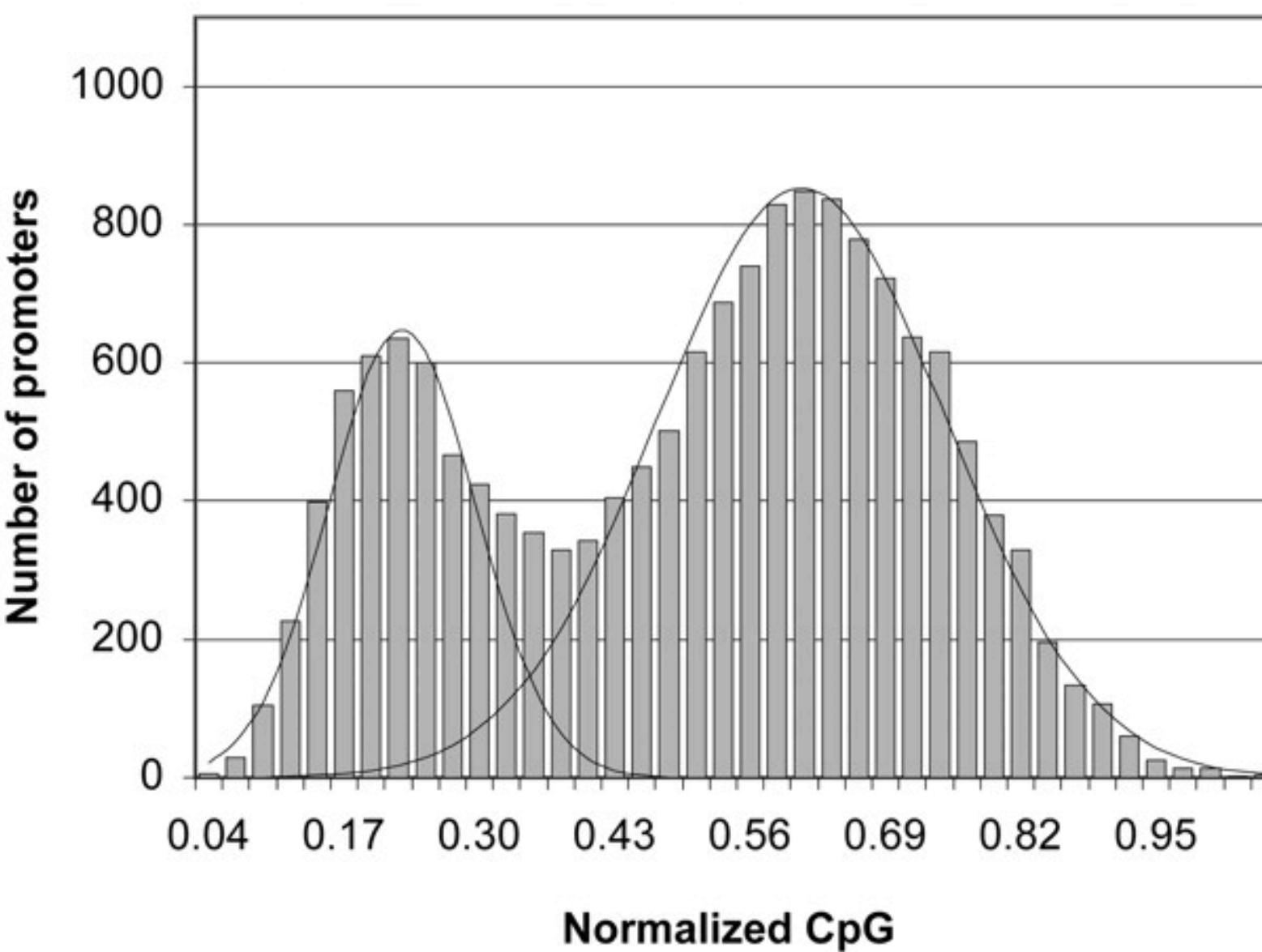


# More Complex Example



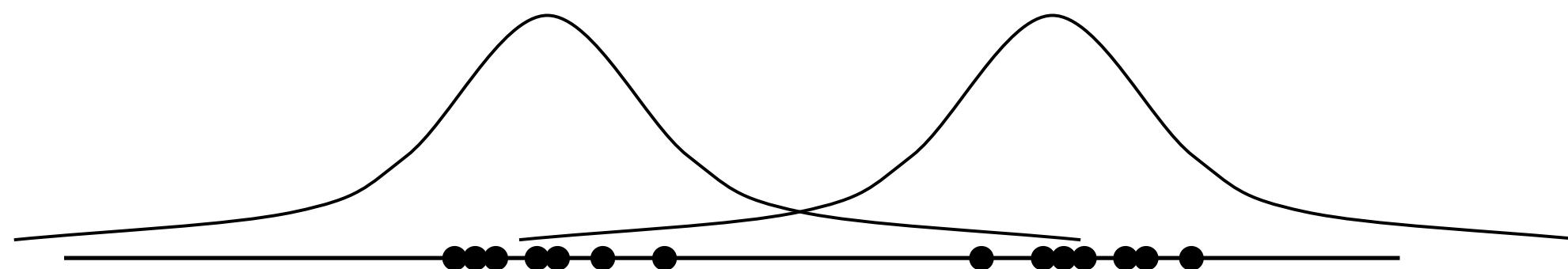
(A modeling decision, not a math problem...,  
but if later, what math?)

## A Real Example: CpG content of human gene promoters



"A genome-wide analysis of CpG dinucleotides in the human genome distinguishes two distinct classes of promoters" Saxonov, Berg, and Brutlag, PNAS 2006;103:1412-1417

## Gaussian Mixture Models / Model-based Clustering



Parameters  $\theta$

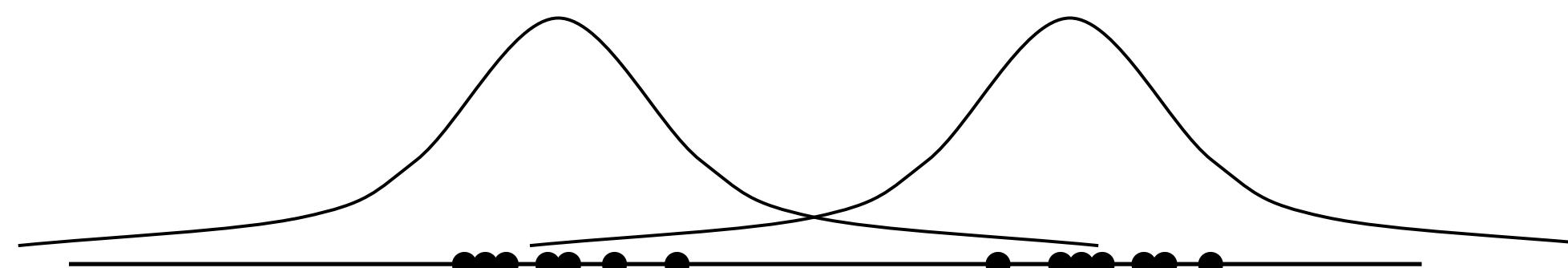
means	$\mu_1$	$\mu_2$
variances	$\sigma_1^2$	$\sigma_2^2$
mixing parameters	$\tau_1$	$\tau_2 = 1 - \tau_1$
P.D.F.	$f(x \mu_1, \sigma_1^2)$	$f(x \mu_2, \sigma_2^2)$

Likelihood

$$\begin{aligned} L(x_1, x_2, \dots, x_n | \mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \tau_1, \tau_2) \\ = \prod_{i=1}^n \sum_{j=1}^2 \tau_j f(x_i | \mu_j, \sigma_j^2) \end{aligned}$$

No  
closed-  
form  
max

## Gaussian Mixture Models / Model-based Clustering



Parameters  $\theta$

means	$\mu_1$	$\mu_2$
variances	$\sigma_1^2$	$\sigma_2^2$
mixing parameters	$\tau_1$	$\tau_2 = 1 - \tau_1$

P.D.F.

$$f(x|\mu_1, \sigma_1^2) \quad f(x|\mu_2, \sigma_2^2)$$

Mixing proportion

Likelihood

$$L(x_1, x_2, \dots, x_n | \mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \tau_1, \tau_2)$$

$$= \left( \prod_{i=1}^n \sum_{j=1}^2 \tau_j f(x_i | \mu_j, \sigma_j^2) \right)$$

Product over data points  
(assumed independent)

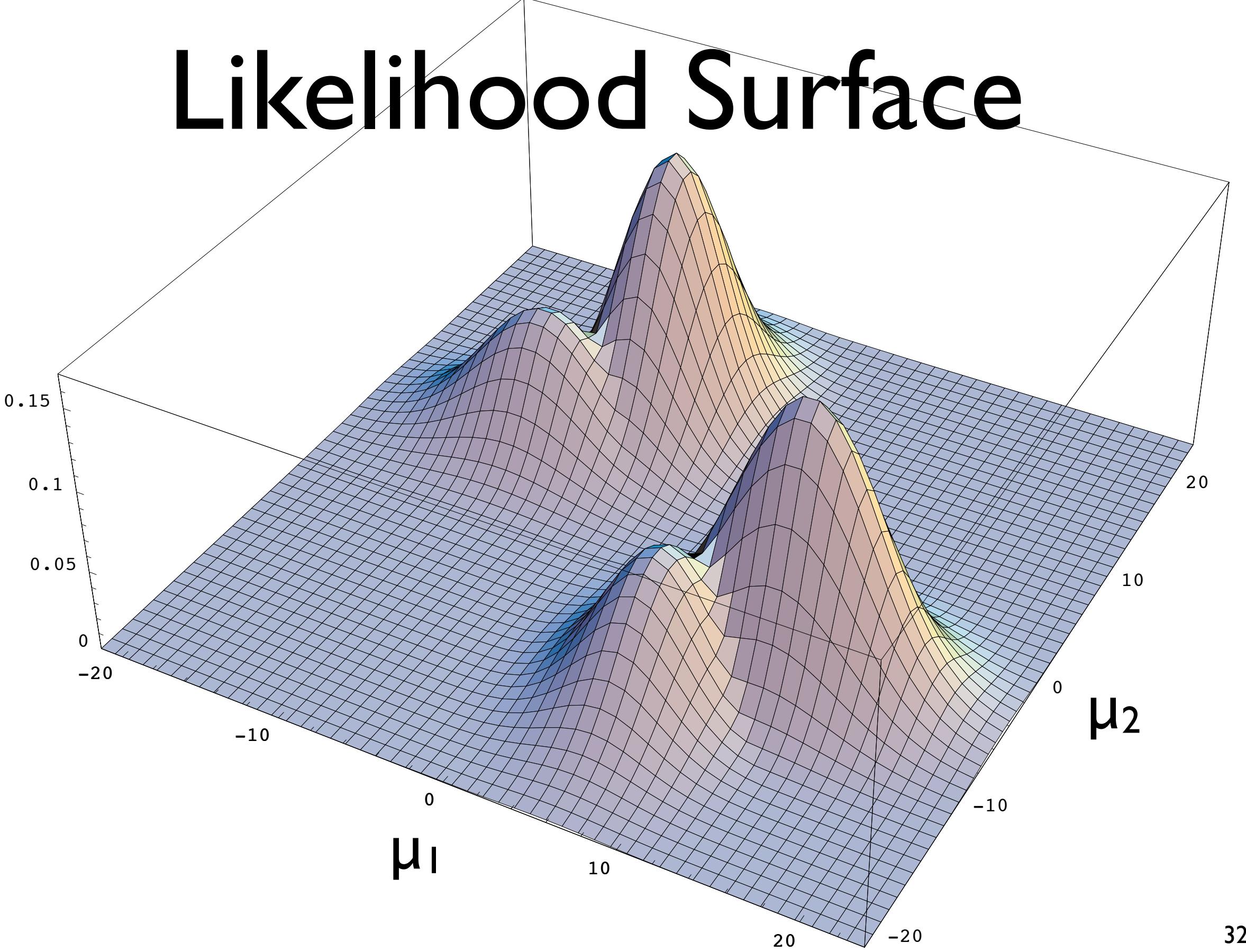
Sum over possible distribution  
of origin

No closed-form max

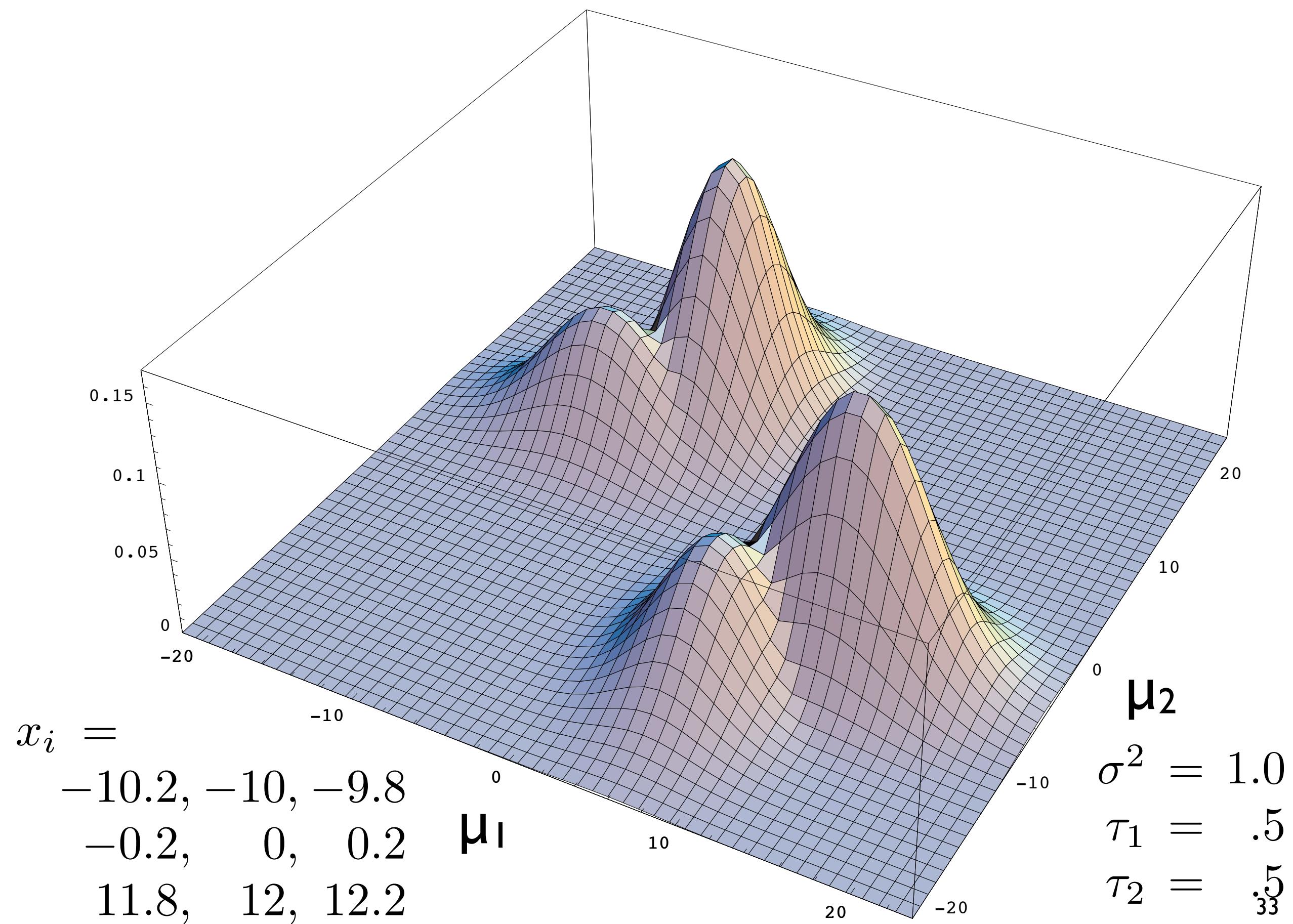
31

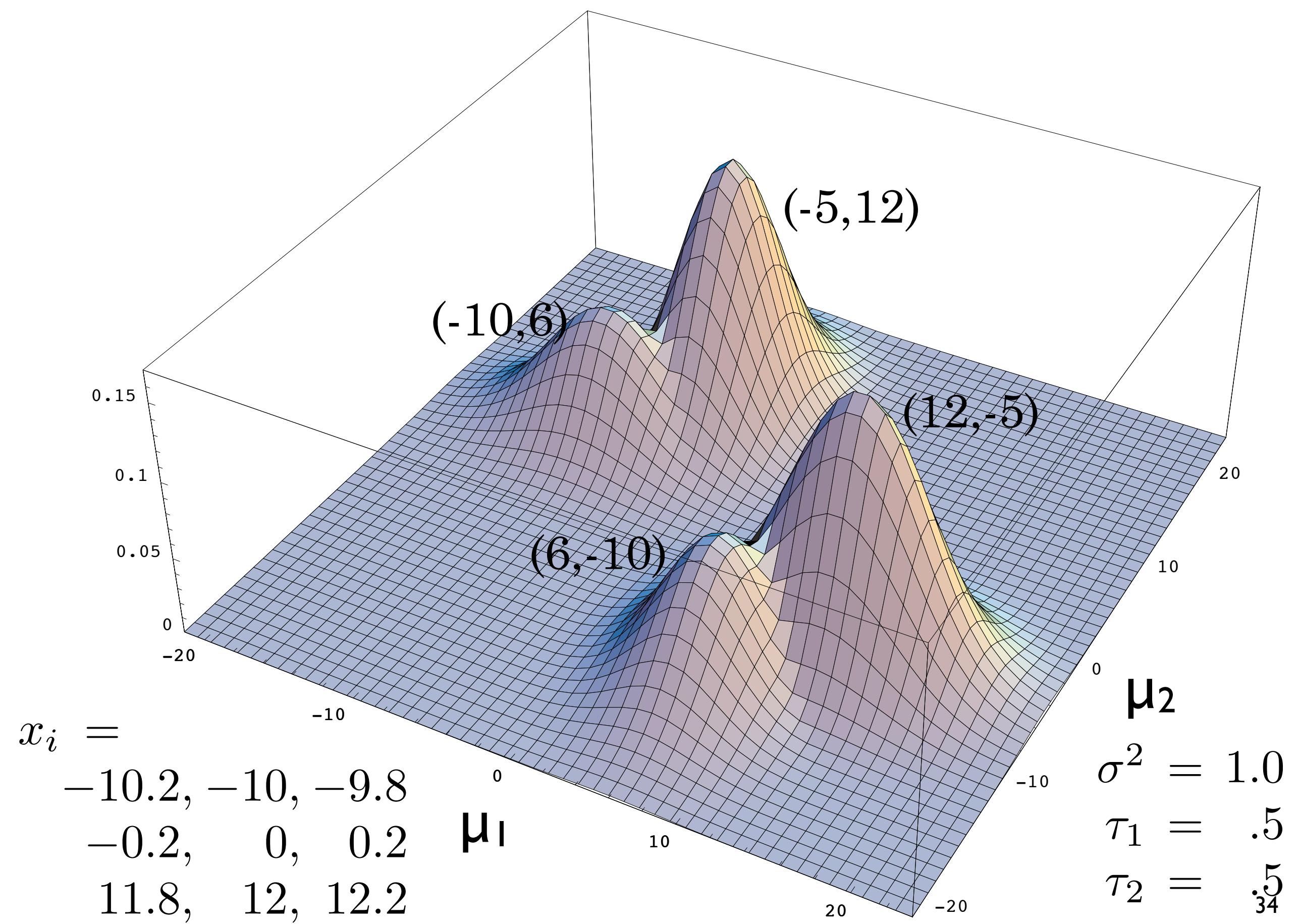
Likelihood of data point given this distribution

# Likelihood Surface



32





# A What-If Puzzle

Likelihood

$$L(x_1, x_2, \dots, x_n | \overbrace{\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \tau_1, \tau_2}^{\theta})$$
$$= \prod_{i=1}^n \sum_{j=1}^2 \tau_j f(x_i | \mu_j, \sigma_j^2)$$

Messy: no closed form solution known for  
finding  $\theta$  maximizing L

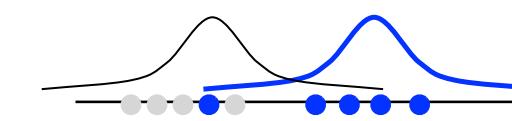
But *what if we  
knew the  
hidden data?*

$$z_{ij} = \begin{cases} 1 & \text{if } x_i \text{ drawn from } f_j \\ 0 & \text{otherwise} \end{cases}$$

# EM as Egg vs Chicken

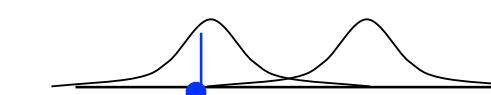
*IF  $z_{ij}$  known, could estimate parameters  $\theta$*

E.g., only points in cluster 2 influence  $\mu_2, \sigma_2$



*IF parameters  $\theta$  known, could estimate  $z_{ij}$*

E.g., if  $|x_i - \mu_1|/\sigma_1 \ll |x_i - \mu_2|/\sigma_2$ , then  $z_{i1} \gg z_{i2}$



**But we know neither; (optimistically) iterate:**

E: calculate expected  $z_{ij}$ , given parameters

M: calc “MLE” of parameters, given  $E(z_{ij})$

Overall, a clever “hill-climbing” strategy

# Simple Version: “Classification EM”

If  $z_{ij} < .5$ , pretend it's 0;  $z_{ij} > .5$ , pretend it's 1

i.e., *classify* points as component 0 or 1

Now recalc  $\theta$ , assuming that partition

Then recalc  $z_{ij}$ , assuming that  $\theta$

Then re-recalc  $\theta$ , assuming new  $z_{ij}$ , etc., etc.

“Full EM” is a bit more involved, but this is the crux.

# Full EM

$x_i$ 's are known;  $\theta$  unknown. Goal is to find MLE  $\theta$  of:

$$L(x_1, \dots, x_n \mid \theta) \quad (\text{hidden data likelihood})$$

Would be easy *if*  $z_{ij}$ 's were known, i.e., consider:

$$L(x_1, \dots, x_n, z_{11}, z_{12}, \dots, z_{n2} \mid \theta) \quad (\text{complete data likelihood})$$

But  $z_{ij}$ 's aren't known.

Instead, maximize *expected* likelihood of visible data

$$E(L(x_1, \dots, x_n, z_{11}, z_{12}, \dots, z_{n2} \mid \theta)),$$

where expectation is over distribution of hidden data ( $z_{ij}$ 's)

# The E-step:

Find  $E(Z_{ij})$ , i.e.  $P(Z_{ij}=1)$

Assume  $\theta$  known & fixed

A (B): the event that  $x_i$  was drawn from  $f_1$  ( $f_2$ )

D: the observed datum  $x_i$

Expected value of  $z_{i1}$  is  $P(A|D)$

$$P(A|D) = \frac{P(D|A)P(A)}{P(D)}$$

$$\begin{aligned} P(D) &= P(D|A)P(A) + P(D|B)P(B) \\ &= f_1(x_i|\theta_1)\tau_1 + f_2(x_i|\theta_2)\tau_2 \end{aligned}$$

Repeat  
for  
each  
 $x_i$

# Complete Data Likelihood

Recall:

$$z_{1j} = \begin{cases} 1 & \text{if } x_1 \text{ drawn from } f_j \\ 0 & \text{otherwise} \end{cases}$$

so, correspondingly,

$$L(x_1, z_{1j} | \theta) = \begin{cases} \tau_1 f_1(x_1 | \theta) & \text{if } z_{11} = 1 \\ \tau_2 f_2(x_1 | \theta) & \text{otherwise} \end{cases}$$

Formulas with “if’s” are messy; can we blend more smoothly?

Yes, many possibilities. Idea 1:

$$L(x_1, z_{1j} | \theta) = z_{11} \cdot \tau_1 f_1(x_1 | \theta) + z_{12} \cdot \tau_2 f_2(x_1 | \theta)$$

Idea 2 (Better):

$$L(x_1, z_{1j} | \theta) = (\tau_1 f_1(x_1 | \theta))^{z_{11}} \cdot (\tau_2 f_2(x_1 | \theta))^{z_{12}}$$

# Complete Data Likelihood

Recall:

$$z_{1j} = \begin{cases} 1 & \text{if } x_1 \text{ drawn from } f_j \\ 0 & \text{otherwise} \end{cases}$$

so, correspondingly,

$$L(x_1, z_{1j} | \theta) = \begin{cases} \tau_1 f_1(x_1 | \theta) & \text{if } z_{11} = 1 \\ \tau_2 f_2(x_1 | \theta) & \text{otherwise} \end{cases}$$

Formulas with "if's" are messy; can we blend more smoothly?

Yes, many possibilities. Idea 1:

$$L(x_1, z_{1j} | \theta) = z_{11} \cdot \tau_1 f_1(x_1 | \theta) + z_{12} \cdot \tau_2 f_2(x_1 | \theta)$$

Idea 2 (Better):

$$L(x_1, z_{1j} | \theta) = \frac{(\tau_1 f_1(x_1 | \theta))^{z_{11}} \cdot (\tau_2 f_2(x_1 | \theta))^{z_{12}}}{\uparrow}$$

40

Why is this better? How will this behave differently when we take the log?

# M-step:

## Find $\theta$ maximizing $E(\log(\text{Likelihood}))$

(For simplicity, assume  $\sigma_1 = \sigma_2 = \sigma; \tau_1 = \tau_2 = .5 = \tau$ )

$$L(\vec{x}, \vec{z} | \theta) = \prod_{1 \leq i \leq n} \frac{\tau}{\sqrt{2\pi\sigma^2}} \exp \left( - \sum_{1 \leq j \leq 2} z_{ij} \frac{(x_i - \mu_j)^2}{2\sigma^2} \right)$$

$$E[\log L(\vec{x}, \vec{z} | \theta)] = E \left[ \sum_{1 \leq i \leq n} \left( \log \tau - \frac{1}{2} \log 2\pi\sigma^2 - \sum_{1 \leq j \leq 2} z_{ij} \frac{(x_i - \mu_j)^2}{2\sigma^2} \right) \right]$$

$$= \sum_{1 \leq i \leq n} \left( \log \tau - \frac{1}{2} \log 2\pi\sigma^2 - \sum_{1 \leq j \leq 2} E[z_{ij}] \frac{(x_i - \mu_j)^2}{2\sigma^2} \right)$$

Find  $\theta$  maximizing this as before, using  $E[z_{ij}]$  found in E-step. Result:

$\mu_j = \sum_{i=1}^n E[z_{ij}]x_i / \sum_{i=1}^n E[z_{ij}]$

(intuit: avg, weighted by subpop prob)

# 2 Component Mixture

$$\sigma_1 = \sigma_2 = 1; \tau = 0.5$$

		<b>mu1</b>	-20.00		-6.00		-5.00		-4.99
		<b>mu2</b>	6.00		0.00		3.75		3.75
<b>x1</b>	<b>-6</b>	<b>z11</b>		5.11E-12		1.00E+00		1.00E+00	
<b>x2</b>	<b>-5</b>	<b>z21</b>		2.61E-23		1.00E+00		1.00E+00	
<b>x3</b>	<b>-4</b>	<b>z31</b>		1.33E-34		9.98E-01		1.00E+00	
<b>x4</b>	<b>0</b>	<b>z41</b>		9.09E-80		1.52E-08		4.11E-03	
<b>x5</b>	<b>4</b>	<b>z51</b>		6.19E-125		5.75E-19		2.64E-18	
<b>x6</b>	<b>5</b>	<b>z61</b>		3.16E-136		1.43E-21		4.20E-22	
<b>x7</b>	<b>6</b>	<b>z71</b>		1.62E-147		3.53E-24		6.69E-26	

Essentially converged in 2 iterations

# Applications

**Clustering is a remarkably successful exploratory data analysis tool**

- Web-search, information retrieval, gene-expression, ...

- Model-based approach above is one of the leading ways to do it

**Gaussian mixture models widely used**

- With many components, empirically match arbitrary distribution

- Often well-justified, due to “hidden parameters” driving the visible data

**EM is extremely widely used for “hidden-data” problems**

- Hidden Markov Models

# EM Summary

Fundamentally a maximum likelihood parameter estimation problem

Useful if hidden data, and if analysis is more tractable when 0/1 hidden data z known

Iterate:

E-step: estimate  $E(z)$  for each  $z$ , given  $\theta$

M-step: estimate  $\theta$  maximizing  $E(\log \text{likelihood})$   
given  $E(z)$  [where “ $E(\log L)$ ” is wrt random  $z \sim E(z) = p(z=1)$ ]

# EM Issues

Under mild assumptions, EM is guaranteed to increase likelihood with every E-M iteration, hence will *converge*.

But it may converge to a *local*, not global, max.  
(Recall the 4-bump surface...)

Issue is intrinsic (probably), since EM is often applied to problems (including clustering, above) that are *NP-hard*

Nevertheless, widely used, often effective

# Aside: Maximum Likelihood Est. and the EM Algorithm

**End of slides on MLE & EM taken from the UW CSE 312 Web\***

# A probabilistic view of RNA-Seq quantification

$$\Pr\{\mathcal{F} \mid \boldsymbol{\eta}, \mathcal{T}\} = \prod_{j=1}^N \Pr\{f_j \mid \boldsymbol{\eta}, \mathcal{T}\}$$

assumes  
independence  
of fragments

nucleotide  
fractions

known  
transcriptome

observed  
fragments  
(reads)

Prob. of selecting  
 $t_i$  given  $\boldsymbol{\eta}$

Depends on  
abundance  
estimate

Prob. of generating  
fragment  $f_j$  given that it originates from  $t_i$

Independent of  
abundance  
estimate

$$= \prod_{j=1}^N \sum_{i=1}^M \Pr\{t_i \mid \boldsymbol{\eta}\} \cdot \Pr\{f_j \mid t_i, z_{ji} = 1\}$$

We want to find the values of  $\boldsymbol{\eta}$  that **maximize** this probability.  
We can do this (at least locally) using the EM algorithm.

# A probabilistic view of RNA-Seq quantification

$$\Pr\{\mathcal{F} \mid \boldsymbol{\eta}, \mathcal{T}\} = \prod_{j=1}^N \Pr\{f_j \mid \boldsymbol{\eta}, \mathcal{T}\}$$

assumes  
independence  
of fragments

nucleotide  
fractions

known  
transcriptome

observed  
fragments  
(reads)

Prob. of selecting  
 $t_i$  given  $\boldsymbol{\eta}$

Depends on  
abundance  
estimate

Prob. of generating  
fragment  $f_j$  given that it originates from  $t_i$

Independent of  
abundance  
estimate

$$= \prod_{j=1}^N \sum_{i=1}^M \Pr\{t_i \mid \boldsymbol{\eta}\} \cdot \boxed{\Pr\{f_j \mid t_i, z_{ji} = 1\}}$$

We want to find the values of  $\boldsymbol{\eta}$  that **maximize** this probability.  
We can do this (at least locally) using the EM algorithm.

# A probabilistic view of RNA-Seq quantification

$$\Pr\{\mathcal{F} \mid \boldsymbol{\eta}, \mathcal{T}\} = \prod_{j=1}^N \Pr\{f_j \mid \boldsymbol{\eta}, \mathcal{T}\}$$

assumes  
independence  
of fragments

nucleotide  
fractions

known  
transcriptome

observed  
fragments  
(reads)

We can safely truncate  $\Pr\{t_i \mid \boldsymbol{\eta}\}$   
to 0 for transcripts where a  
fragment doesn't map/align.

$$= \prod_{j=1}^N \left( \sum_{i=1}^M \Pr\{t_i \mid \boldsymbol{\eta}\} \cdot \Pr\{f_j \mid t_i, z_{ji} = 1\} \right)$$

Prob. of selecting  
 $t_i$  given  $\boldsymbol{\eta}$

Depends on  
abundance  
estimate

Prob. of generating  
fragment  $f_j$  given that it originates from  $t_i$

Independent of  
abundance  
estimate

We want to find the values of  $\boldsymbol{\eta}$  that **maximize** this probability.  
We can do this (at least locally) using the EM algorithm.

# A probabilistic view of RNA-Seq quantification

E-step: (what is the “soft assignment” of each read to the transcripts where it aligns)

$$E_{Z|\mathcal{F},\eta^{(t)}}[Z_{nij}] = P(Z_{nij} = 1 | \mathcal{F}, \eta^{(t)}) = \frac{(\eta_i^{(t)} / \ell_i) P(f_n | Z_{nij} = 1)}{\sum_{i',j'} (\eta_{i'}^{(t)} / \ell_{i'}) P(f_n | Z_{ni'j'} = 1)}$$

M-step: Given these soft assignments, how abundant is each transcript?

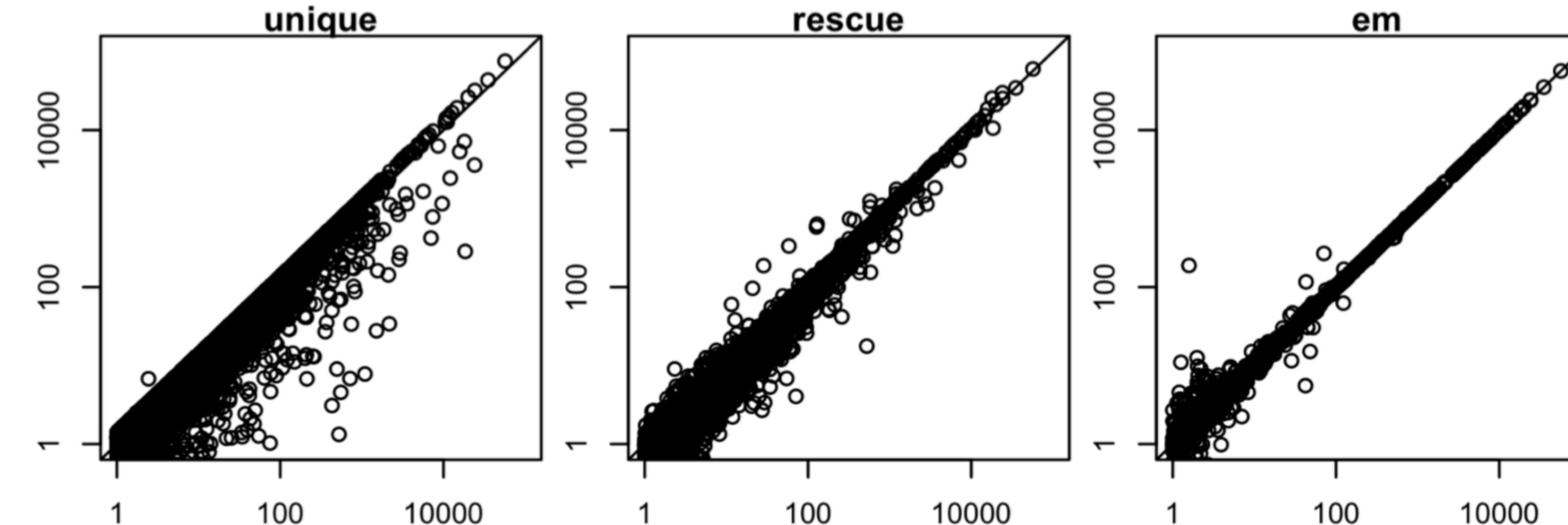
$$\eta_i^{(t+1)} = \frac{E_{Z|\mathcal{F},\eta^{(t)}} [C_i]}{N},$$

$$\text{where } C_i = \sum_{n,j} Z_{nij}$$

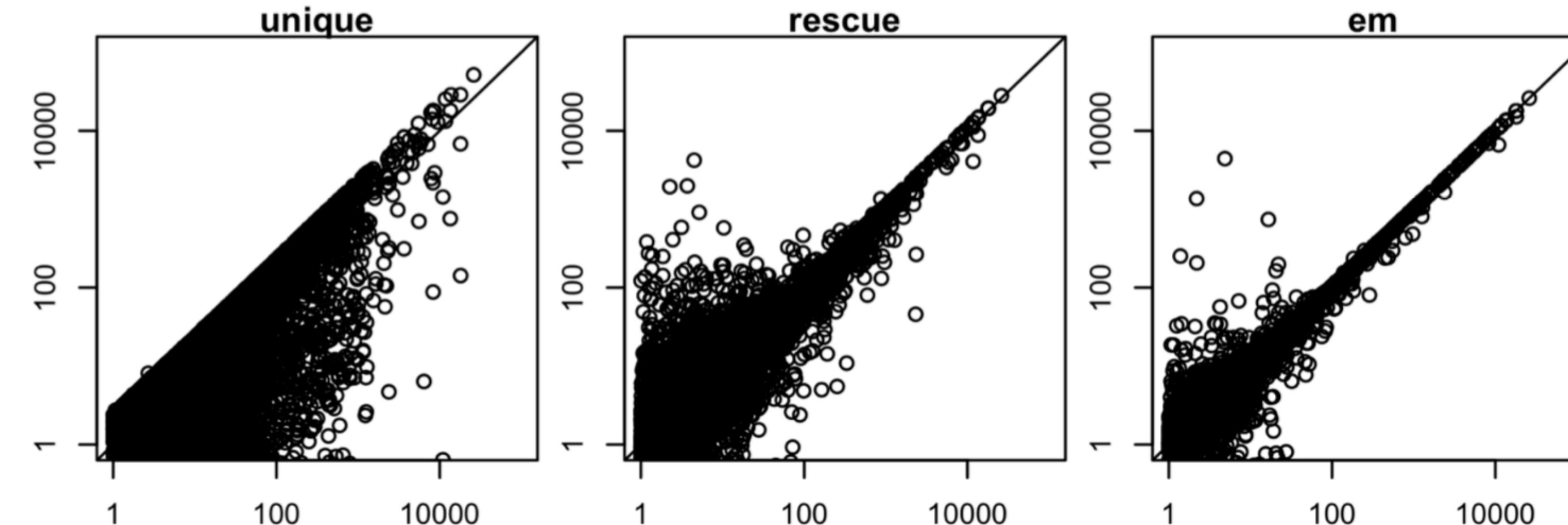
This approach is quite effective. Unfortunately, it's also quite slow.

# Gene expression estimation accuracy in simulated data

Mouse liver



Maize



# A probabilistic view of RNA-Seq quantification

We want to find the values of  $\eta$  that **maximize** this probability.  
We can do this (at least locally) using the EM algorithm.

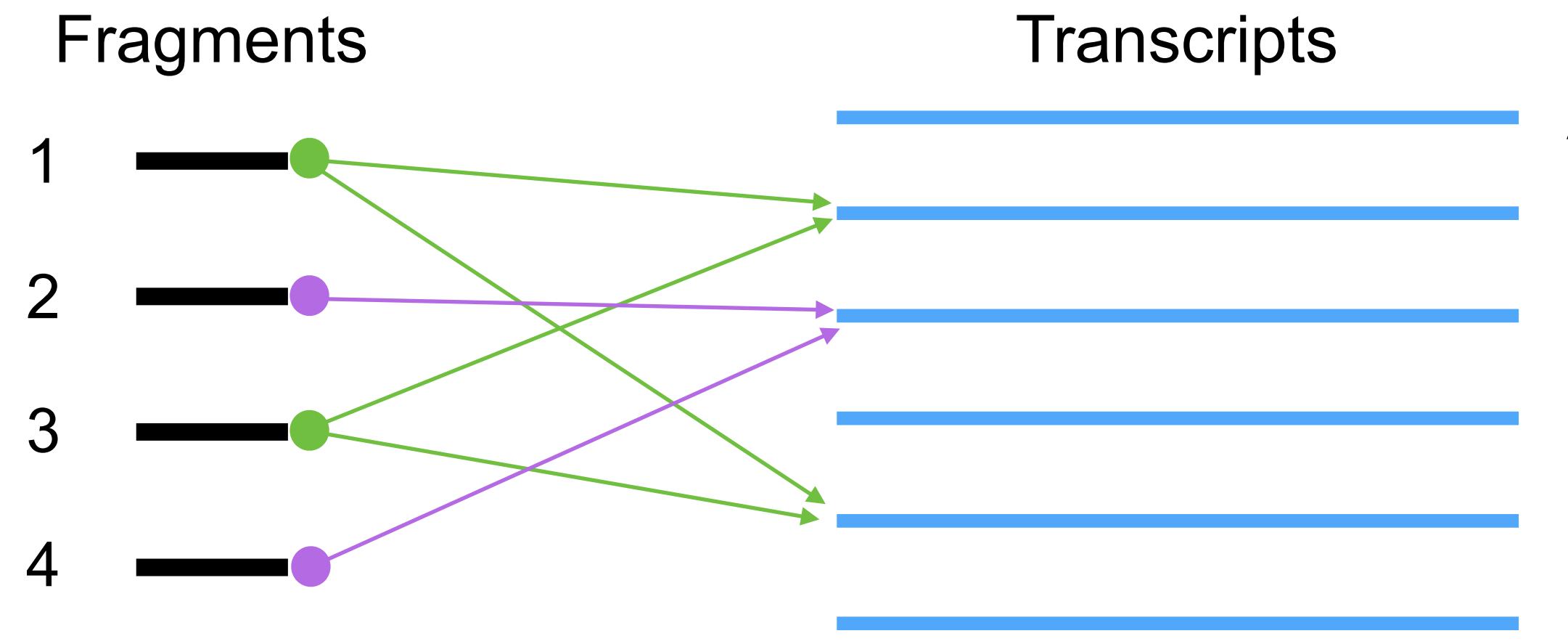
**but**

This leads to an iterative EM algorithm where each *iteration* scales in the total number of **alignments** in the sample (typically on the order of  $10^7 — 10^8$ ), and typically  $10^2 — 10^3$  **iterations**

$$\mathcal{L}(\boldsymbol{\eta}; \mathcal{F}, \mathcal{T}) = \prod_{f \in \mathcal{F}} \sum_{t_i \in \Omega(f)} \Pr(t_i | \boldsymbol{\eta}) \Pr(f | t_i)$$

Set of transcripts where f maps/aligns

# Fragment Equivalence Classes



Reads 1 & 3 both map to transcripts B & E

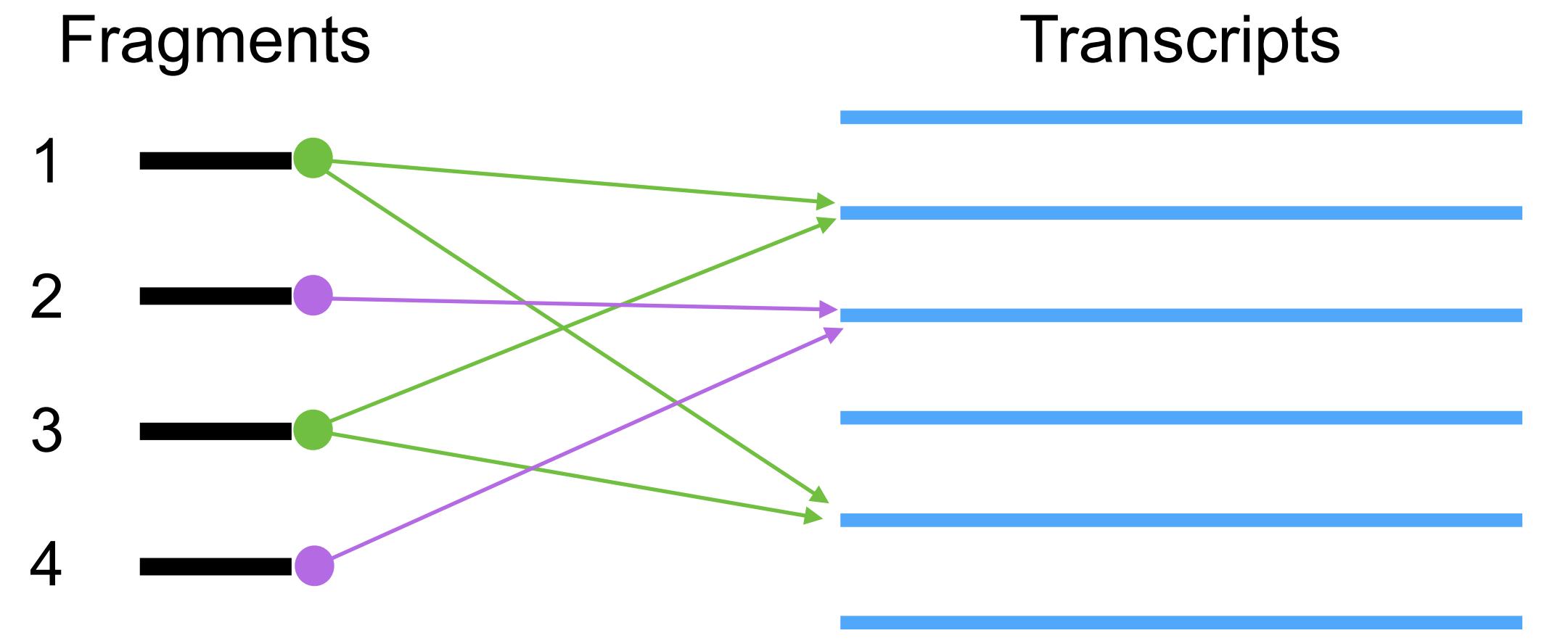
Reads 2 & 4 both map to transcript C

We have 4 reads, but only 2 eq. classes of reads

eq. Label	Count	Aux weights
{B,E}	2	$w^{(B,E)}_B, w^{(B,E)}_E$
{C}	2	$w^{(C)}_C$

This idea goes quite far back in the RNA-seq literature; at least to MMSeq (Turro et al. 2011)

# Fragment Equivalence Classes



Reads 1 & 3 both map to transcripts B & E  
Reads 2 & 4 both map to transcript C

$w_{j|i}$  encodes the “affinity” of class  $j$  to transcript  $i$  according to the model. This is  $P\{f_j | t_i\}$ , aggregated for all fragments in a class.

We have 4 reads, but only 2 eq. classes of reads

eq. Label	Count	Aux weights
{B,E}	2	$w^{(B,E)}_B, w^{(B,E)}_E$
{C}	2	$w^{(C)}_C$

This idea goes quite far back in the RNA-seq literature; at least to MMSeq (Turro et al. 2011)

# The number of equivalence classes is **small**

	Yeast	Human	Chicken
# contigs	7353	107,389	335,377
# samples	6	6	8
Total (paired-end) reads	~36,000,000	~116,000,000	~181,402,780
Avg # eq. classes (across samples)	5197	100,535	222,216

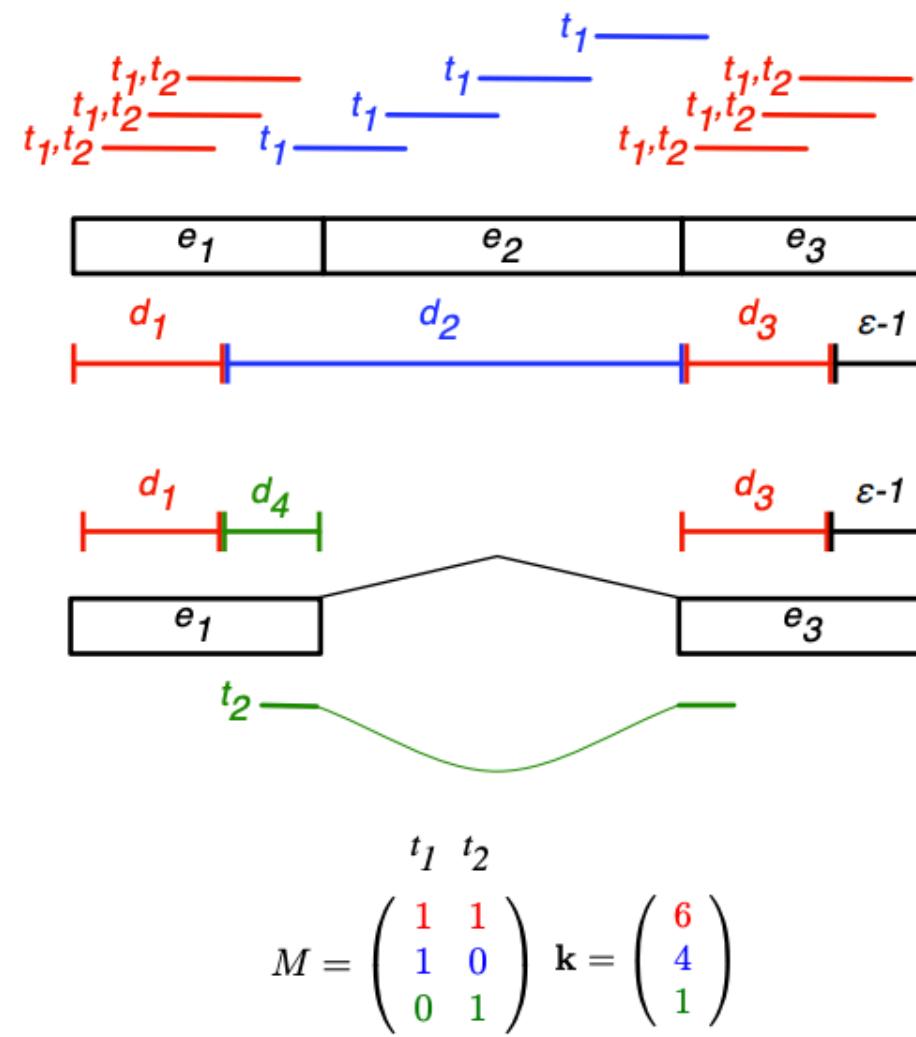
The **# of equivalence classes grows with the complexity of the transcriptome** — independent of the # of sequence fragments.

Typically, **two or more orders of magnitude** fewer equivalence classes than sequenced fragments.

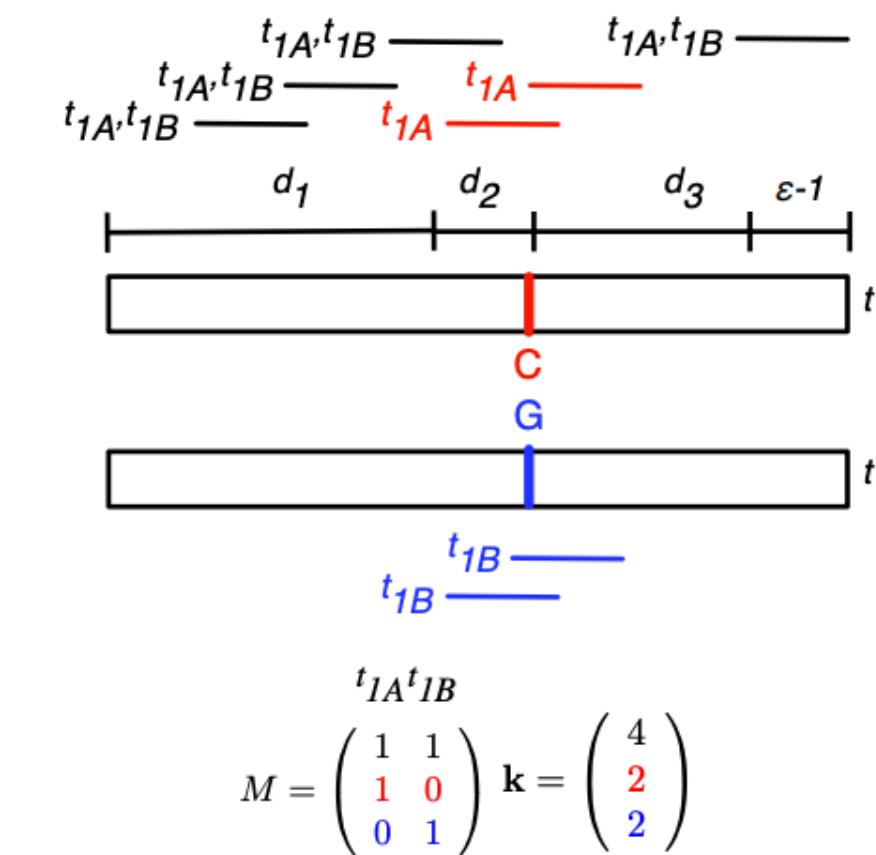
The offline **inference** algorithm **scales in # of fragment equivalence classes**.

# This naturally handles different types of multi-mapping without having to rely on the annotation

(a)



(b)



# This lets us approximate the likelihood efficiently

Approximate this:

$$\mathcal{L}(\boldsymbol{\eta}; \mathcal{F}) = \prod_{f_j \in \mathcal{F}} \sum_{i=1}^M \Pr(t_i \mid \boldsymbol{\eta}) \Pr(f_j \mid t_i)$$

sum over all alignments of fragment

product over all fragments

with this:

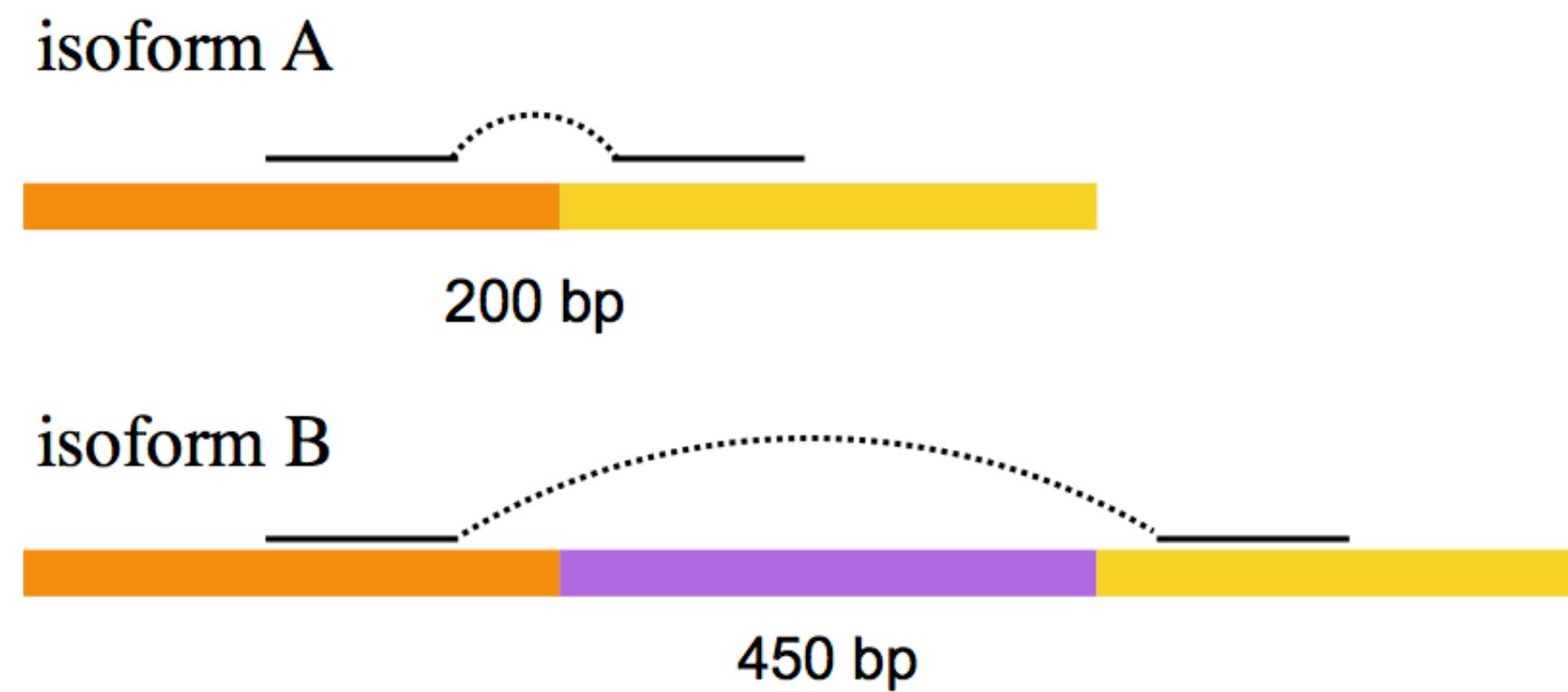
$$\mathcal{L}(\boldsymbol{\eta}; \mathcal{F}) \approx \prod_{\mathcal{F}^q \in \mathcal{C}} \left( \sum_{\langle i, t_i \rangle \in \Omega(\mathcal{F}^q)} \Pr(t_i \mid \boldsymbol{\eta}) \cdot \Pr(f \mid \mathcal{F}^q, t_i) \right)^{N^q}$$

sum over all transcripts labeling this eq. class

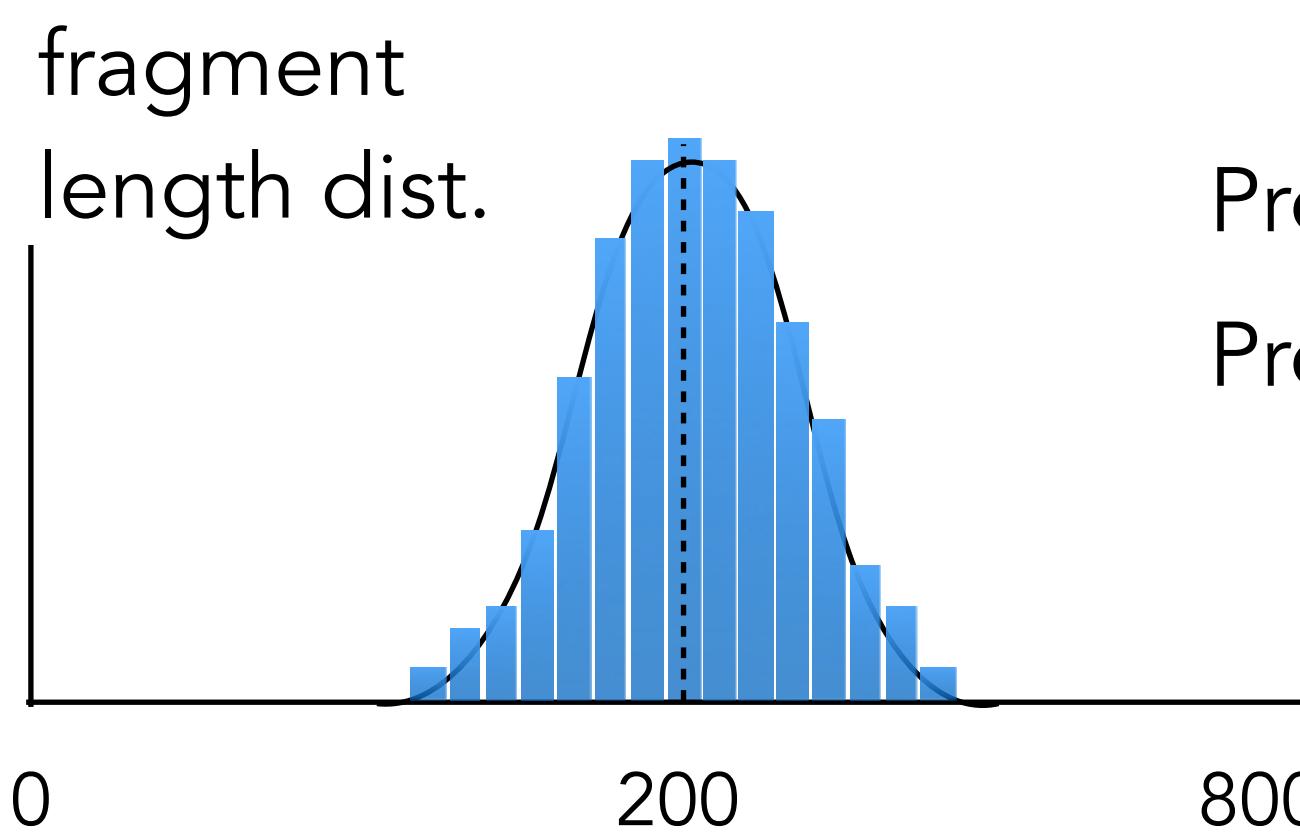
product over all equivalence classes

# Why might $\text{Pr}(f_j \mid t_i)$ matter?

Consider the following scenario:



Conditional probabilities can provide valuable information about origin of a fragment! **Potentially different for each transcript/fragment pair.**



Prob of observing a fragment of size  $\sim 200$  is **large**  
Prob of observing a fragment of size  $\sim 450$  is **small**

**Many terms can be considered in a general “fragment-transcript agreement” model<sup>1</sup>. e.g. position, orientation, alignment path etc.**

<sup>1</sup> “Salmon provides fast and bias-aware quantification of transcript expression”, Nature Methods 2017

# Optimizing the objective

Estimation of background bias models  
Recomputation of effective lengths  
Offline algorithm runs until convergence

offline inference  
[EM or VBEM]

our ML objective has a simple, **closed-form update rule** in terms of our eq. classes

$$\alpha_i^{u+1} = \sum_{\mathcal{F}^q \in \mathcal{C}} N^q \left( \frac{\alpha_i^u w_i^q}{\sum_{\langle k, t_k \rangle \in \Omega(\mathcal{F}^q)} \alpha_k^u w_k^q} \right)$$

estimated read count from transcript i at iteration u+1

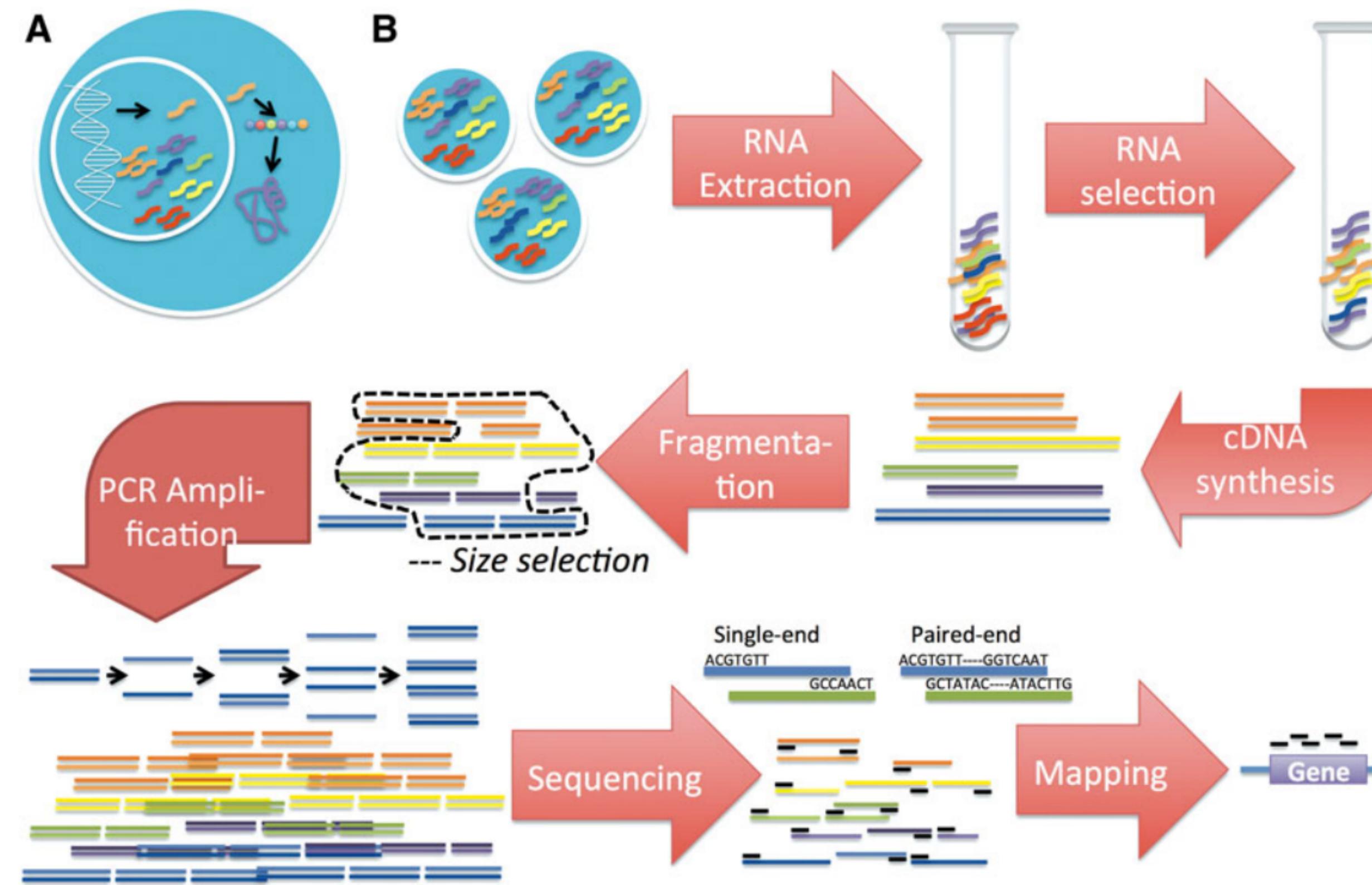
count of eq. class j

weight of  $t_i$  in eq. class q

$$\hat{\eta}_i = \frac{\alpha_i}{\sum_j \alpha_j}$$

we also provide the *option* to use a **variational Bayesian** objective instead

# Actual RNA-seq protocols are a bit more “involved”



There is **substantial** potential for biases and deviations from the *basic* model — indeed, we see quite a few.

# Biases abound in RNA-seq data

Biases in prep & sequencing can have a significant effect on the fragments we see:

Fragment gc-bias<sup>1</sup>—

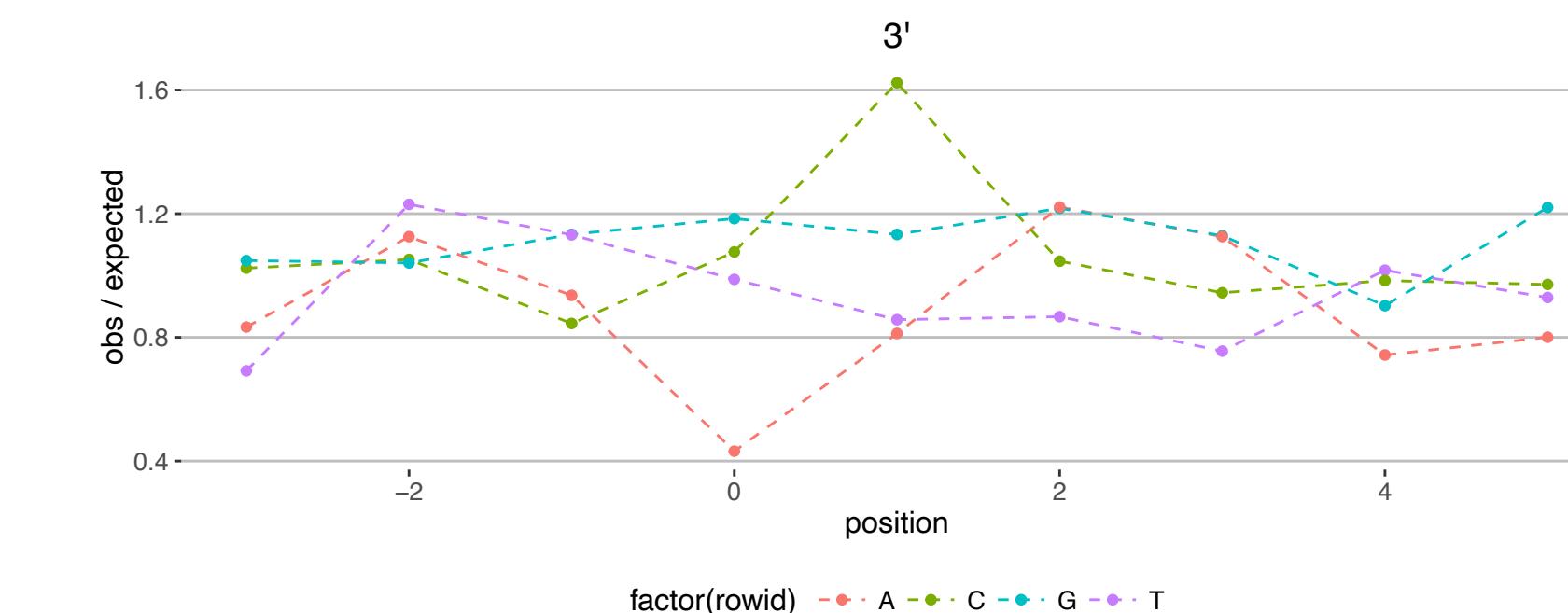
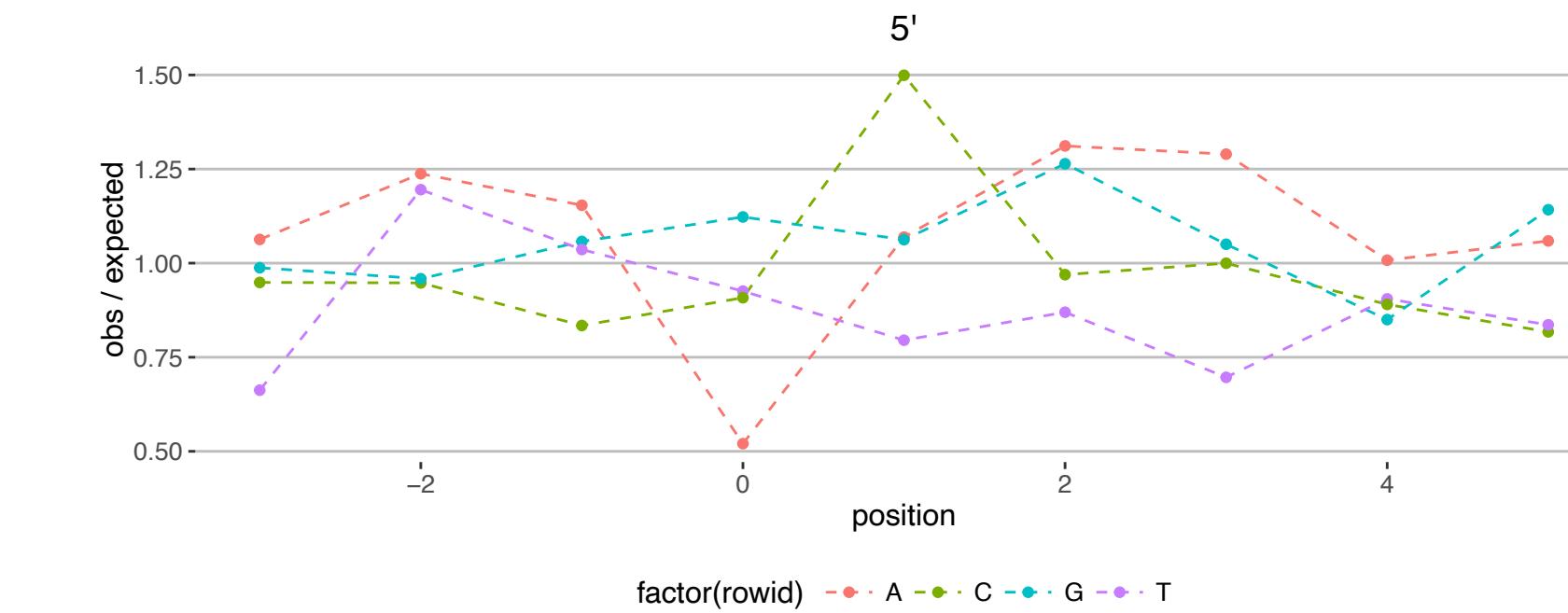
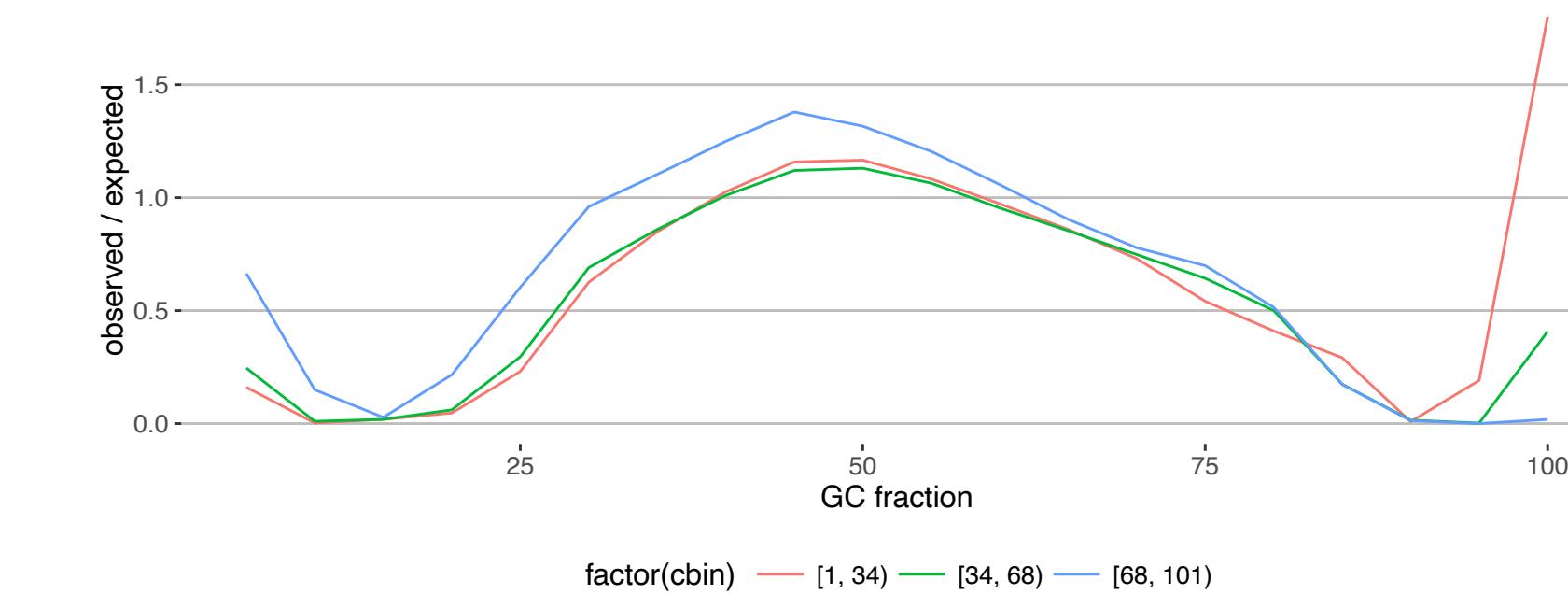
The GC-content of the fragment affects the likelihood of sequencing

Sequence-specific bias<sup>2</sup>—

sequences surrounding fragment affect the likelihood of sequencing

Positional bias<sup>2</sup>—

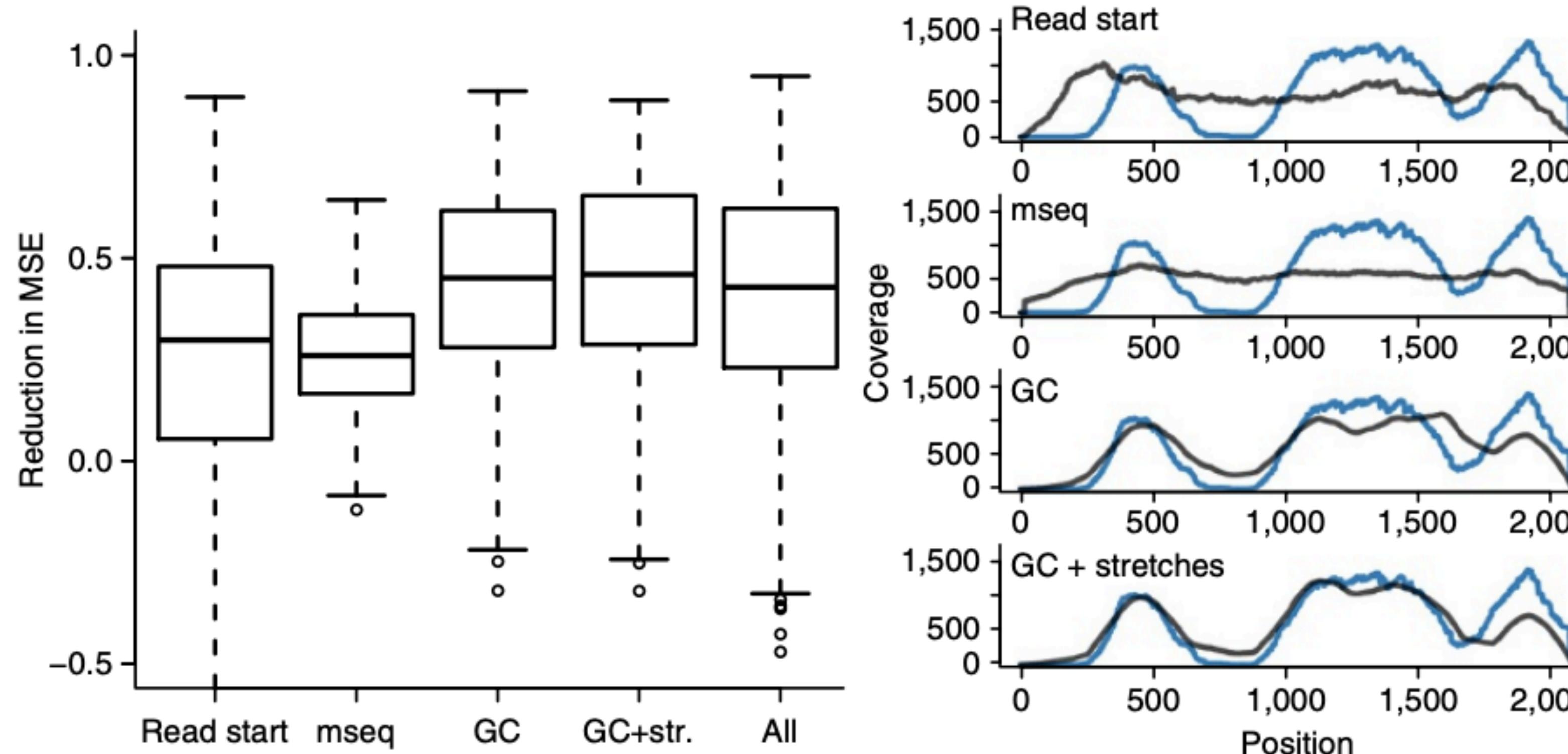
fragments sequenced non-uniformly across the body of a transcript



1:Love, Michael I., John B. Hogenesch, and Rafael A. Irizarry. "Modeling of RNA-seq fragment sequence bias reduces systematic errors in transcript abundance estimation." bioRxiv (2015): 025767.

2:Roberts, Adam, et al. "Improving RNA-Seq expression estimates by correcting for fragment bias." Genome biology 12.3 (2011): 1.

# Biases abound in RNA-seq data



**Fragment GC-bias is often the most extreme**

# Biases abound in RNA-seq data

**Basic idea (1):** Modify the “effective length” of a transcript to account for changes in the sampling probability. This leads to changes in soft-assignment in EM -> changes in TPM.

Fragment gc-bias<sup>1</sup>—

The GC-content of the fragment affects the likelihood of sequencing

**Basic idea (2):** The effective length of a transcript is the sum of the bias terms at each position across a transcript. The bias term at a given position is simply the (observed / expected) sampling probability.

Positional bias<sup>2</sup>—

The trick is how to define “expected” given only biased data.

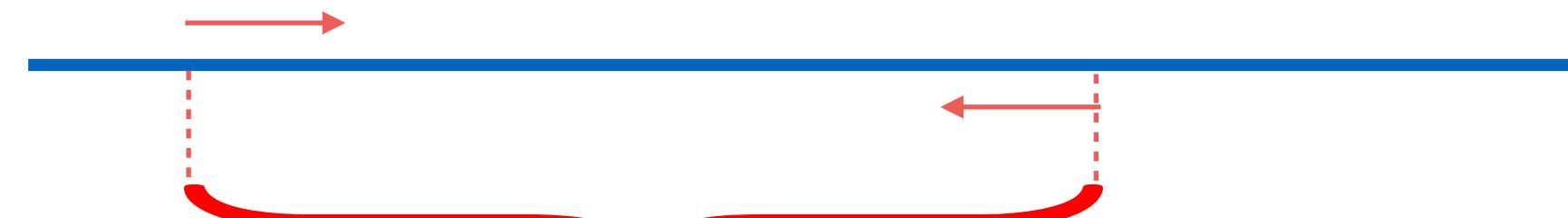
# Bias Modeling

Bias correction works by adjusting the effective lengths of the transcripts:  
The effective length becomes the sum of the per-base biases

$$\tilde{\ell}'_i = \sum_{j=1}^{j \leq \ell_i} \sum_{k=1}^{k \leq f_i(j, L)} \frac{b_{gc^+}(t_i, j, j+k)}{b_{gc^-}(t_i, j, j+k)} \cdot \frac{b_{s^+}^{5'}(t_i, j)}{b_{s^-}^{5'}(t_i, j)} \cdot \frac{b_{s^+}^{3'}(t_i, j+k)}{b_{s^-}^{3'}(t_i, j+k)} \cdot \frac{b_{p^+}^{5'}(t_i, j+k)}{b_{p^-}^{5'}(t_i, j+k)} \cdot \frac{b_{p^+}^{3'}(t_i, j+k)}{b_{p^-}^{3'}(t_i, j+k)} \cdot \Pr\{X=j\}$$

Fragment GC bias model:

Density of fragments with specific GC content,  
**conditioned** on GC fraction at read start/end



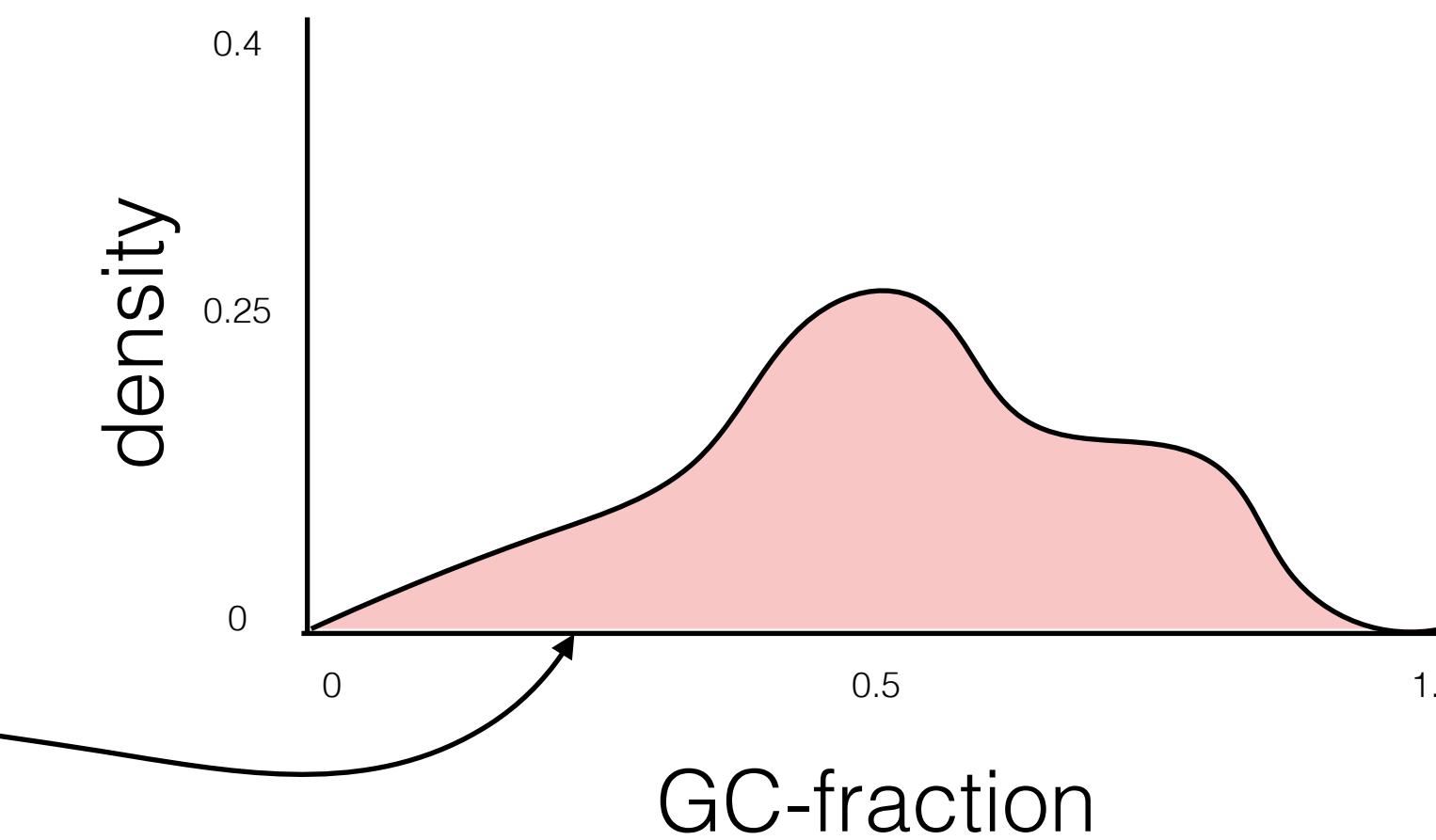
GC-fraction of fragment

**Foreground:**

Observed

**Background:**

Expected given est. abundances



# Bias Modeling

Bias correction works by adjusting the effective lengths of the transcripts:  
The effective length becomes the sum of the per-base biases

$$\tilde{\ell}'_i = \sum_{j=1}^{j \leq \ell_i} \sum_{k=1}^{k \leq f_i(j, L)} \frac{b_{gc^+}(t_i, j, j+k)}{b_{gc^-}(t_i, j, j+k)} \cdot \frac{b_{s^+}^{5'}(t_i, j)}{b_{s^-}^{5'}(t_i, j)} \cdot \frac{b_{s^+}^{3'}(t_i, j+k)}{b_{s^-}^{3'}(t_i, j+k)} \cdot \frac{b_{p^+}^{5'}(t_i, j+k)}{b_{p^-}^{5'}(t_i, j+k)} \cdot \frac{b_{p^+}^{3'}(t_i, j+k)}{b_{p^-}^{3'}(t_i, j+k)} \cdot \Pr\{X = j\}$$

Seq-specific bias model\*:

VLMM for the 10bp window surrounding the 5'  
read start site and the 3' read start site

**Foreground:**

Observed

**Background:**

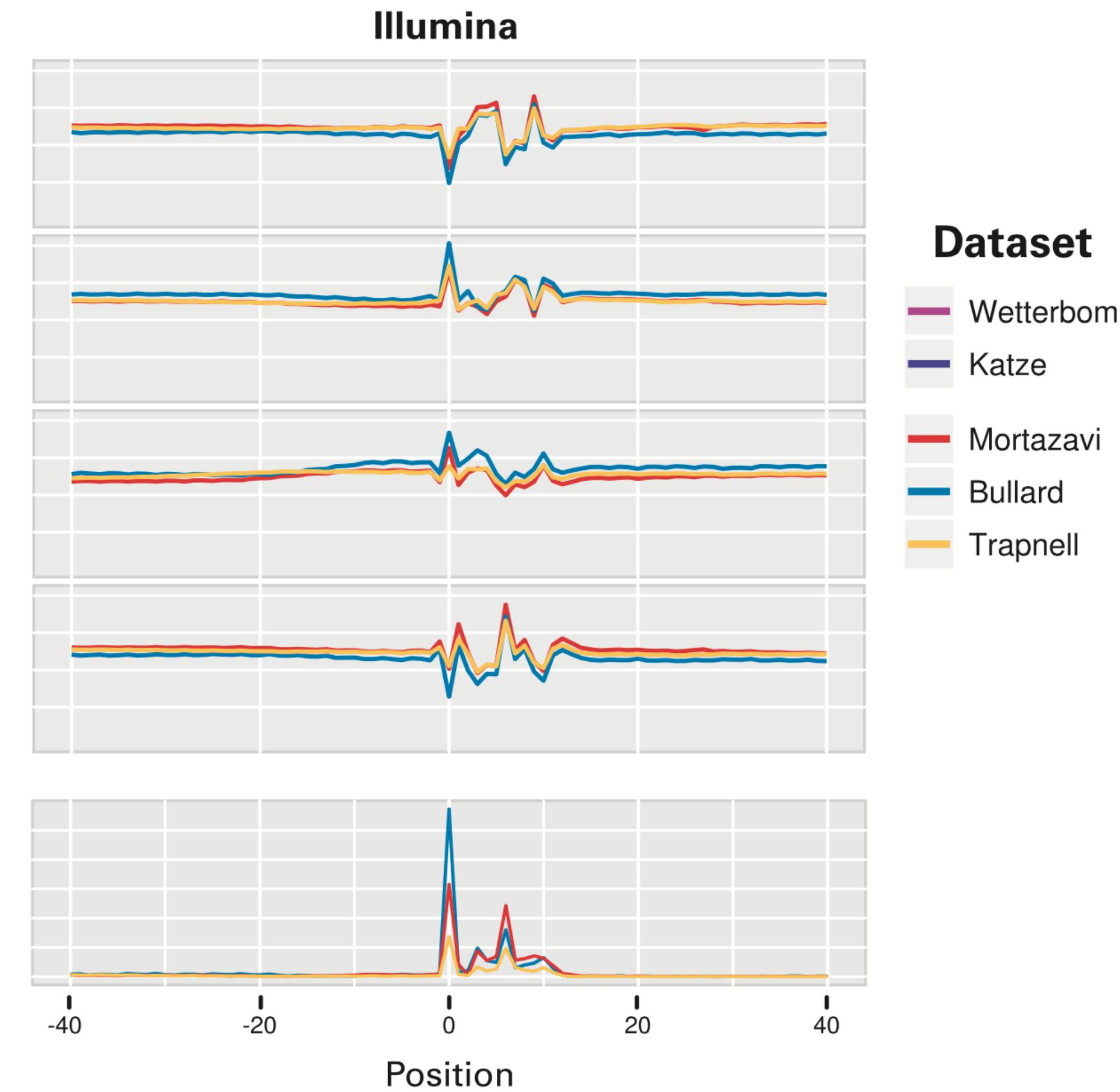
Expected given est. abundances



Add this sequence to training set with weight =  
 $P\{f \mid t_i\}$

Same, but independent  
model for 3' end

# Priming bias is sample & sequence-specific



# Bias Modeling

Bias correction works by adjusting the effective lengths of the transcripts:  
The effective length becomes the sum of the per-base biases

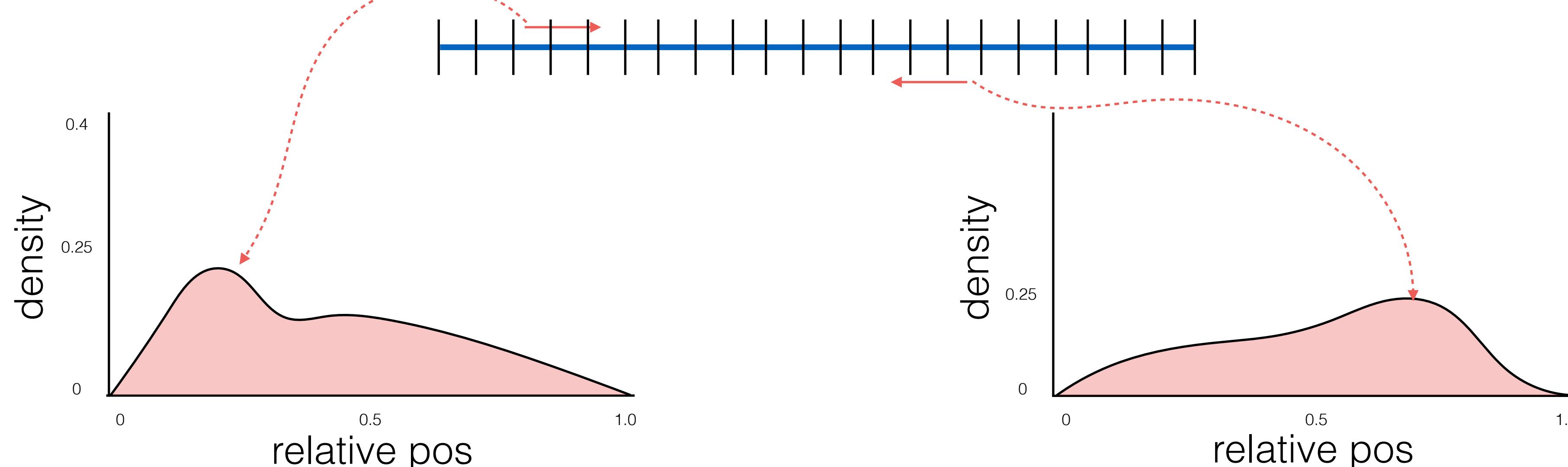
$$\tilde{\ell}'_i = \sum_{j=1}^{j \leq \ell_i} \sum_{k=1}^{k \leq f_i(j, L)} \frac{b_{gc^+}(t_i, j, j+k)}{b_{gc^-}(t_i, j, j+k)} \cdot \frac{b_{s^+}^{5'}(t_i, j)}{b_{s^-}^{5'}(t_i, j)} \cdot \frac{b_{s^+}^{3'}(t_i, j+k)}{b_{s^-}^{3'}(t_i, j+k)} \cdot \frac{b_{p^+}^{5'}(t_i, j+k)}{b_{p^-}^{5'}(t_i, j+k)} \cdot \frac{b_{p^+}^{3'}(t_i, j+k)}{b_{p^-}^{3'}(t_i, j+k)} \cdot \Pr\{X=j\}$$

Position bias model\*:

Density of 5' and 3' read start positions —  
different models for transcripts of different length

**Foreground:**  
Observed

**Background:**  
Expected given est. abundances



\*Roberts, Adam, et al. "Improving RNA-Seq expression estimates by correcting for fragment bias." Genome biology 12.3 (2011): 1.

# **Estimating Posterior Uncertainty**

## One “issue” with maximum likelihood (ML)

The generative statistical model is a principled and elegant way to represent the RNA-seq process.

It can be optimized efficiently using e.g. the EM / VBEM algorithm.

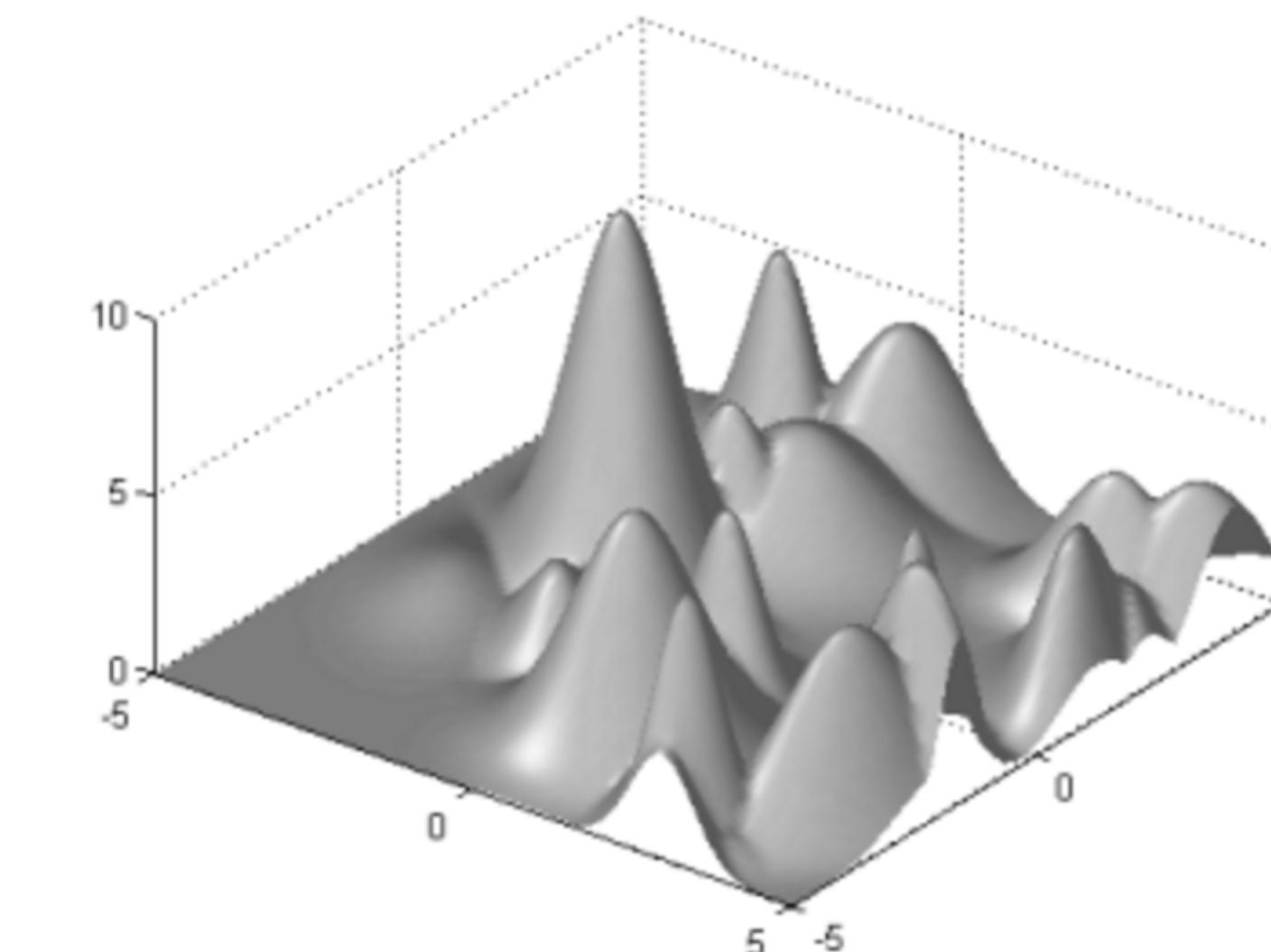
**but**, these efficient optimization algorithms return “point estimates” of the abundances. That is, there is no notion of how *certain* we are in the computed abundance of transcript.

# One “issue” with maximum likelihood (ML)

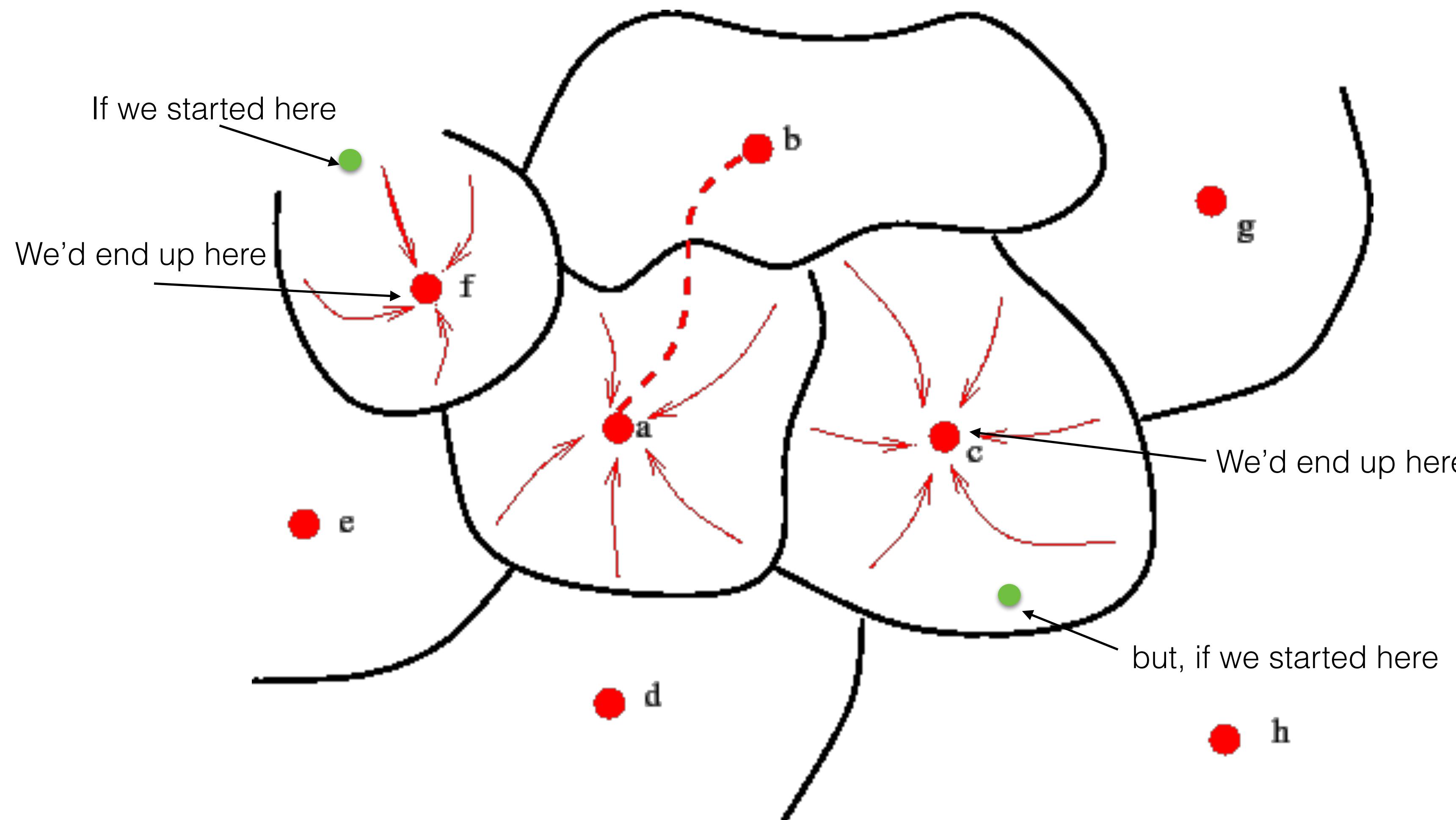
There are multiple sources of uncertainty e.g.

- Technical variance : If we sequenced the *exact* same sample again, we'd get a different set of fragments, and, potentially a different solution.
- Uncertainty in inference: We are almost never guaranteed to find a unique, globally optimal result. If we started our algorithm with different initialization parameters, we might get a different result.

We're trying to find the *best* parameters in a space with 10s to 100s of thousands of dimensions!



# One “issue” with maximum likelihood (ML)



# Assessing Uncertainty

There are a few ways to address this “issue”

Do a fully Bayesian inference<sup>1</sup>:

Infer the entire posterior distribution of parameters, not just a ML estimate (e.g. using MCMC) — too slow!

✓ Posterior Gibbs Sampling<sup>2,3</sup>:

Starting from our ML estimate, do MCMC sampling to explore how parameters vary — if our ML estimate is good, this can be made *quite fast*.

✓ Bootstrap Sampling<sup>4</sup>:

Resample (from range-factorized equivalence class counts) with replacement, and re-run the ML estimate for each sample. This can be made reasonably fast.

1: BitSeq (with MCMC) actually does this. It’s very accurate, but very slow. [Glaus, Peter, Antti Honkela, and Magnus Rattray. "Identifying differentially expressed transcripts from RNA-seq data with biological variation." *Bioinformatics* 28.13 (2012): 1721-1728.]

2: RSEM has the ability to do this, and it seems to work well, but each sample scales in the # of reads. [Li, Bo, and Colin N. Dewey. "RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome." *BMC bioinformatics* 12.1 (2011): 1.]

3: MMSEQ can perform Gibbs sampling over shared variables (i.e. equiv classes), producing estimates from the mean of the posterior dist. Turro, Ernest, et al. "Haplotype and isoform specific expression estimation using multi-mapping RNA-seq reads." *Genome biology* 12.2 (2011): 1.

4: IsoDE introduced the idea of bootstrapping counts to assess quantification uncertainty. [Al Seesi, Sahar, et al. "Bootstrap-based differential gene expression analysis for RNA-Seq data with and without replicates." *BMC genomics* 15.8 (2014): 1.], but it was first made practical / fast in kallisto by doing the bootstrapping over equivalence classes.

# A few ways to implement Gibbs Sampling for this problem

The model of MMSeq

$$X_{it} \mid \mu_t \sim Pois(bs_i M_{it} \mu_t), \quad (12)$$

$$\mu_t \sim Gam(\alpha, \beta). \quad (13)$$

The full conditionals are:

$$\{X_{i1}, \dots, X_{it}\} \mid \{\mu_1, \dots, \mu_t\}, k_i \sim Mult\left(k_i, \frac{M_{i1}\mu_1}{\sum_t M_{it}\mu_t}, \dots, \frac{M_{in}\mu_n}{\sum_t M_{it}\mu_t}\right), \quad (14)$$

$$\mu_t \mid \{X_{1t}, \dots, X_{mt}\} \sim Gam\left(\alpha + \sum_i X_{it}, \beta + bl_t\right). \quad (15)$$

Again, the  $s_i$  are not needed as they are absent from the full conditionals.

# A few ways to implement Gibbs Sampling for this problem

## The model of BitSeq

$$P(I_n | \boldsymbol{\theta}, \theta^{act}, R) = \text{Cat}(I_n | \boldsymbol{\phi}_n), \quad (10)$$

$$\phi_{n0} = P(r_n | \text{noise})(1 - \theta^{act}) / Z_n^{(\phi)},$$

$$m \neq 0; \phi_{nm} = P(r_n | I_n) \theta_m \theta^{act} / Z_n^{(\phi)},$$

$$P(\boldsymbol{\theta} | \mathbf{I}, \theta^{act}, R) = \text{Dir}(\boldsymbol{\theta} | (\alpha^{dir} + C_1, \dots, \alpha^{dir} + C_M)), \quad (11)$$

$$P(\theta^{act} | \mathbf{I}, \boldsymbol{\theta}, R) = \text{Beta}(\theta^{act} | \alpha^{act} + N - C_0, \beta^{act} + C_0), \quad (12)$$

$$C_m = \sum_{n=1}^N \delta(I_n = m).$$

# A few ways to implement Gibbs Sampling for this problem

## The model of BitSeq (collapsed sampler)

$$P(I_n|I^{(-n)}, R) = \text{Cat}(I_n|\phi_{\mathbf{n}}^*), \quad (9)$$

$$\phi_{n0}^* = P(r_n|\text{noise})(\beta^{act} + C_0^{(-n)})/Z_n^{(\phi^*)},$$

$$m \neq 0; \phi_{nm}^* = P(r_n|I_n)(\alpha^{act} + C_+^{(-n)}) \frac{(\alpha^{dir} + C_m^{(-n)})}{(M\alpha^{dir} + C_+^{(-n)})}/Z_n^{(\phi^*)},$$

$$C_m^{(-n)} = \sum_{i \neq n} \delta(I_i = m),$$

$$C_+^{(-n)} = \sum_{i \neq n} \delta(I_i > 0),$$

with  $Z_n^{(\phi^*)}$  being a constant normalising  $\phi_{\mathbf{n}}^*$  to sum up to 1, and  $\alpha^{dir} = 1, \alpha^{act} = 2, \beta^{act} = 2$ .

# This uncertainty matters

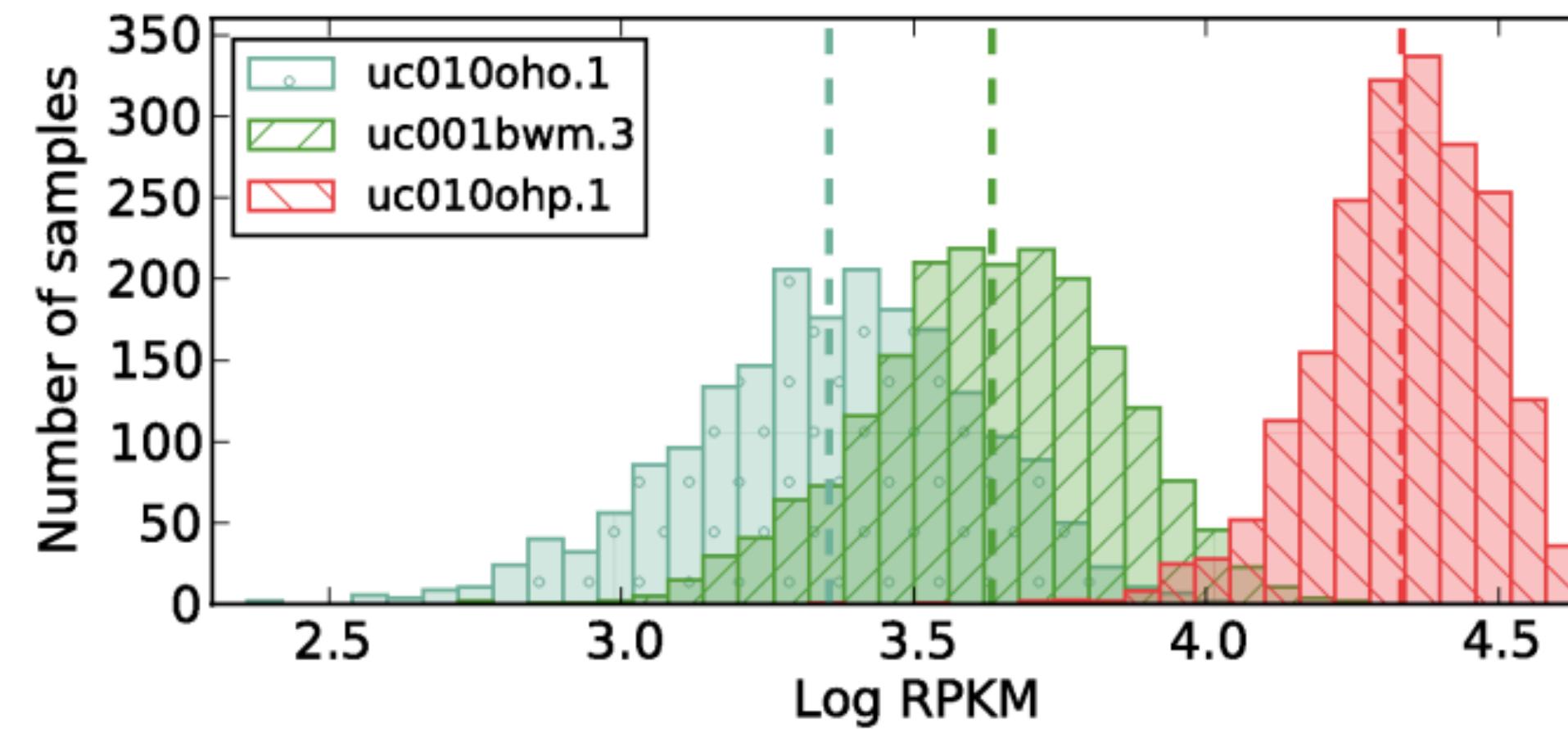
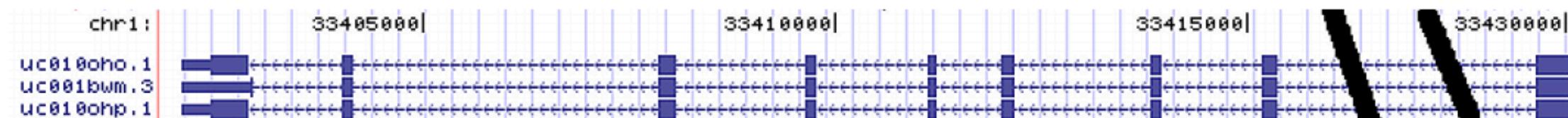
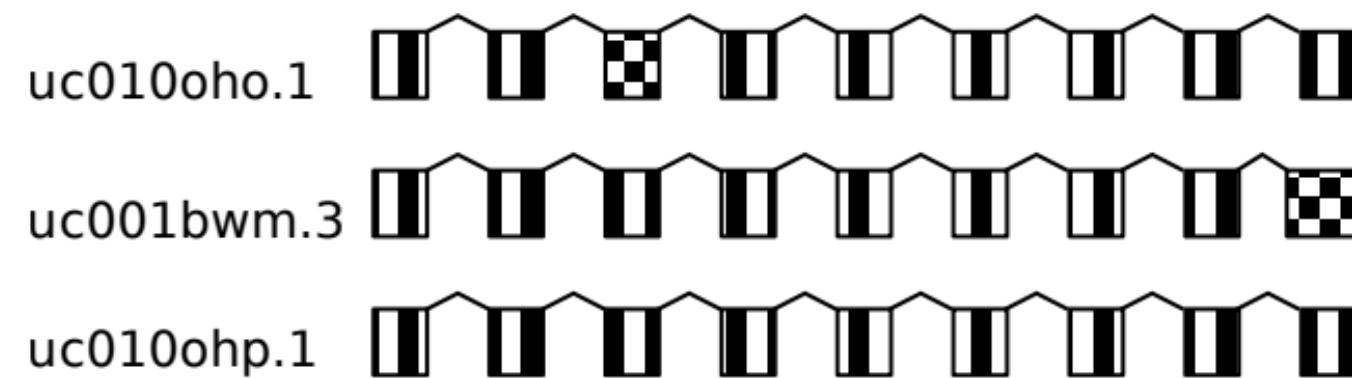


Figure 2.10: **Posterior distribution of expression levels of three transcripts of gene Q6ZMZ0.** The posterior distribution is represented in form of a histogram of expression samples converted into Log RPKM expression measure. The dashed lines mark the mean expression for each transcript.

# This uncertainty matters

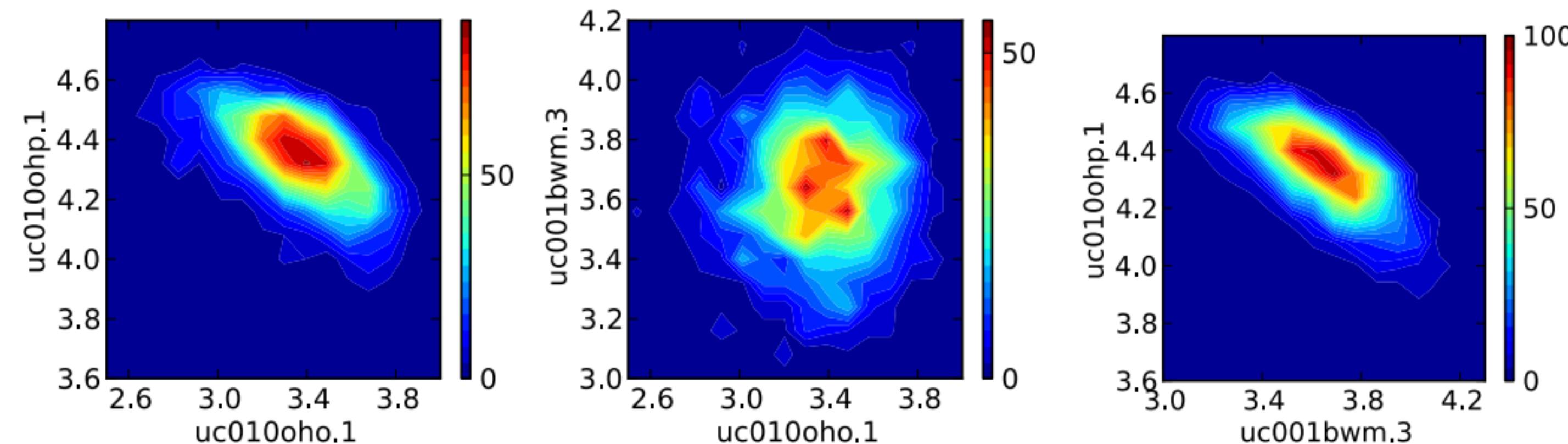


(a) Transcript sequence profile.



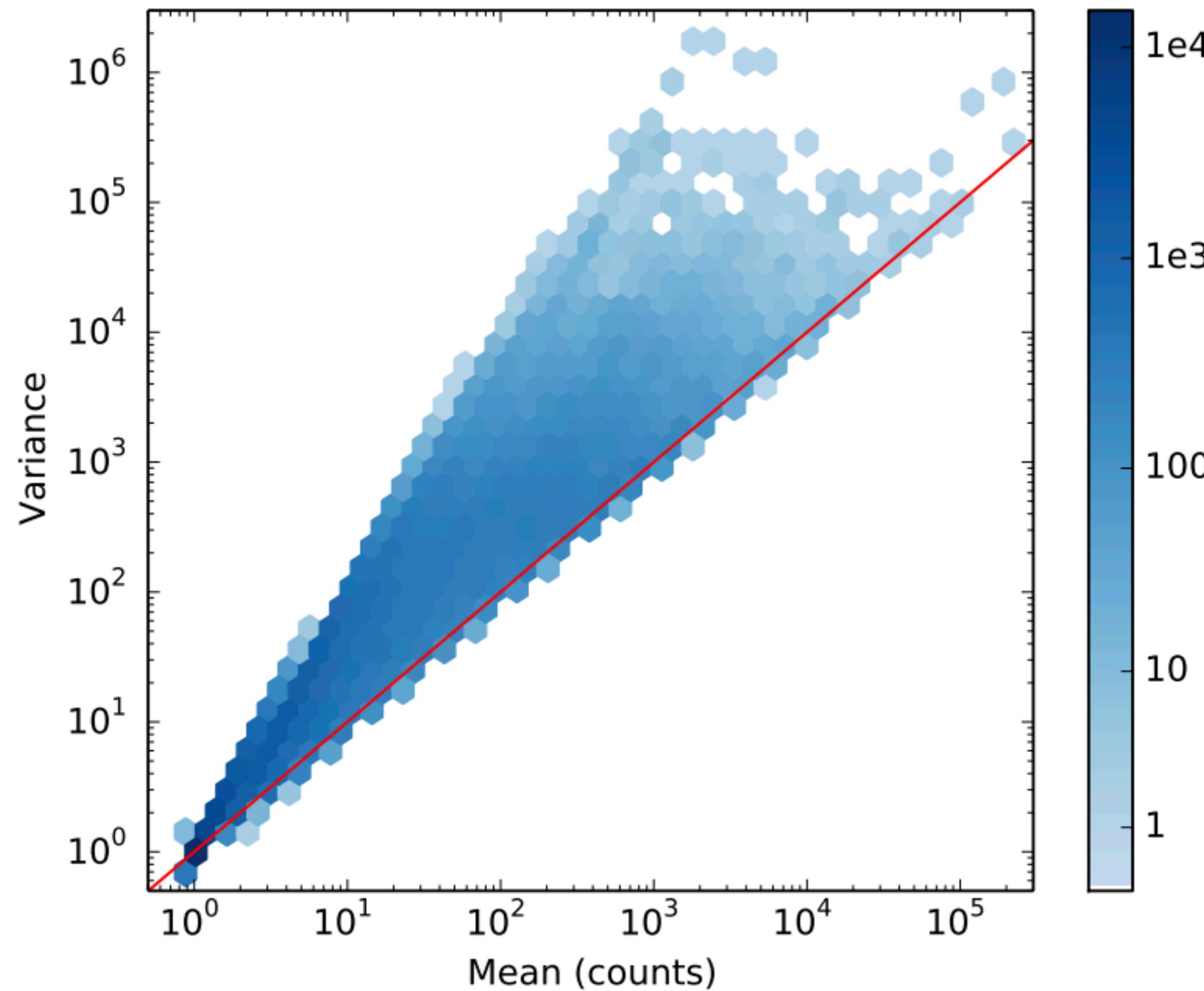
(b) Splice variant model.

Figure 2.12: **Exon model of transcripts of gene Q6ZMZ0.** (a) transcript sequence profile obtained from the UCSC genome browser (Kuhn et al., 2013). In this annotation, transcript uc001bwm.3 has different 3' untranslated region and transcript uc010oho.1 has extra nucleotides at the end of second exon. As the second change cannot be distinguished in the UCSC genome browser diagram, we provide schematic splice variant model highlighting the differences (b).



# This uncertainty matters

We observe considerably increased variance due to read mapping ambiguity



If we know this increased uncertainty, we can propagate it & use it in downstream analysis (differential expression)!