RNA-Seq Analysis and Gene Discovery

BMI/CS 776

www.biostat.wisc.edu/bmi776/

Spring 2022

Daifeng Wang

daifeng.wang@wisc.edu

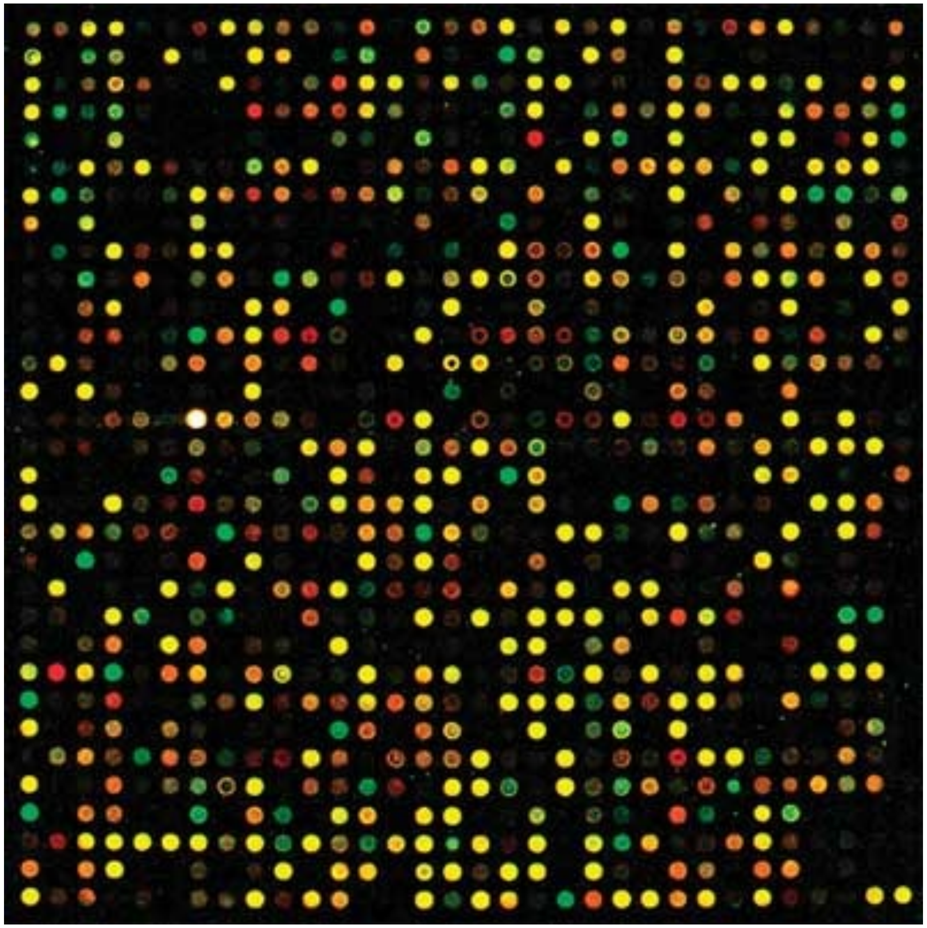
Overview

- RNA-Seq technology
- The RNA-Seq quantification problem
- Interpolated Markov Model
  - Finding bacterial genes

Goals for lecture

- What is RNA-Seq?
- How is RNA-Seq used to measure the abundances of RNAs within cells?
- What probabilistic models and algorithms are used for analyzing RNA-Seq?
- Finding genes

Measuring transcription the old way: microarrays



- Each spot has “probes” for a certain gene
- Probe: a DNA sequence complementary to a certain gene
- Relies on complementary hybridization
- Intensity/color of light from each spot is measurement of the number of transcripts for a certain gene in a sample
- Requires knowledge of gene sequences

Advantages of RNA-Seq over microarrays

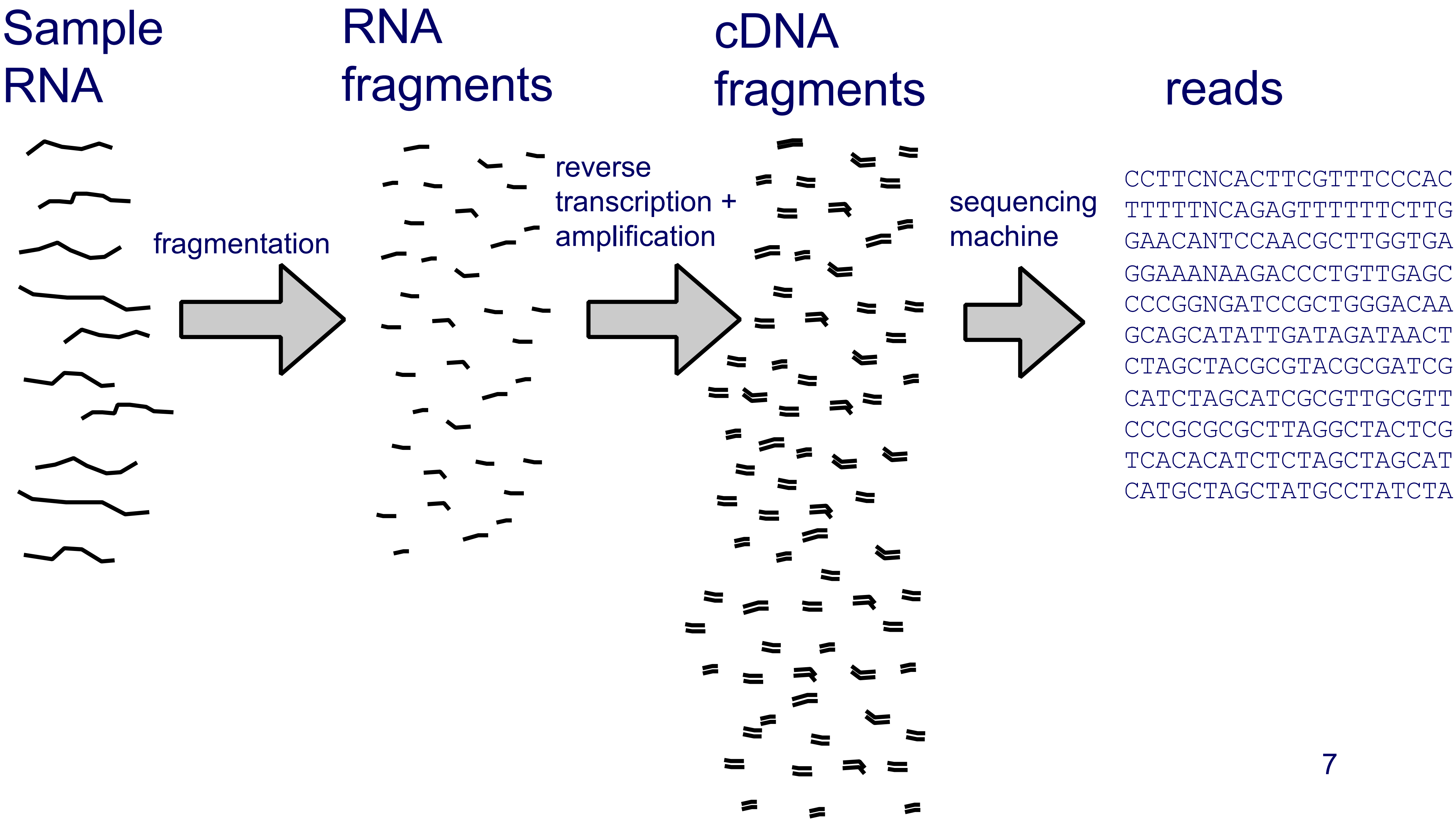
- No reference sequence needed
  - With microarrays, limited to the probes on the chip
- Low background noise
- Large dynamic range
  - 10^5 compared to 10^2 for microarrays
- High technical reproducibility
- Identify novel transcripts and splicing events

RNA-Seq technology

- Leverages rapidly advancing sequencing technology
- Transcriptome analog to whole genome shotgun sequencing
- Two key differences from genome sequencing:

1. Transcripts sequenced at different levels of coverage - expression levels
2. Sequences already known (in many cases) - coverage is measurement

A generic RNA-Seq protocol



Sample RNA

fragmentation

RNA fragments

reverse transcription + amplification

cDNA fragments

sequencing machine

reads

```
CCTTCNCACTTCGTTTCCCAC
TTTTTNCAGAGTTTTTTCTTG
GAACANTCCAACGCTTGGTGA
GGAAANAAGACCCTGTTGAGC
CCCGGNGATCCGCTGGGACAA
GCAGCATATTGATAGATAACT
CTAGCTACGCGTACGCGATCG
CATCTAGCATCGCGTTGCGTT
CCCGCGCGCTTAGGCTACTCG
TCACACATCTCTAGCTAGCAT
CATGCTAGCTATGCCTATCTA
```

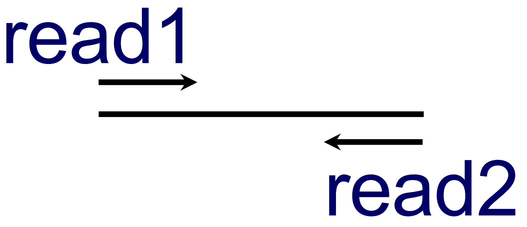
RNA-Seq data: FASTQ format

```
@HWUSI-EAS1789_0001:3:2:1708:1305#0/1
CCTTCNCACTTCGTTTCCCACTTAGCGATAATTTG
+HWUSI-EAS1789_0001:3:2:1708:1305#0/1
VVULVBVYVYZZXZZ\ee[a^b`[a\a[\\a^^^\
@HWUSI-EAS1789_0001:3:2:2062:1304#0/1
TTTTTNCAGAGTTTTTTCTTGAACTGGAAATTTTT
+HWUSI-EAS1789_0001:3:2:2062:1304#0/1
a__[\Bbbb`edeeefd`cc`b]bffff`ffffff
@HWUSI-EAS1789_0001:3:2:3194:1303#0/1
GAACANTCCAACGCTTGGTGAATTCTGCTTCACAA
+HWUSI-EAS1789_0001:3:2:3194:1303#0/1
ZZ[[VBZZY][TWQQZ\ZS\[ZZXV__\OX`a[ZZ
@HWUSI-EAS1789_0001:3:2:3716:1304#0/1
GGAAANAAGACCCTGTTGAGCTTGACTCTAGTCTG
+HWUSI-EAS1789_0001:3:2:3716:1304#0/1
aaXWYBZVTXZX_]XdccdfbbB_\`a\aY_^]LZ^
@HWUSI-EAS1789_0001:3:2:5000:1304#0/1
CCCGGNGATCCGCTGGGACAAGCAGCATATTGATA
+HWUSI-EAS1789_0001:3:2:5000:1304#0/1
aaaaaBeeeeffffehhhhhhggdhhhhahhhadh
```

name
sequence
qualities

read

paired-end reads



read1

read2

1 Illumina HiSeq 2500 lane

~150 million reads

Tasks with RNA-Seq data

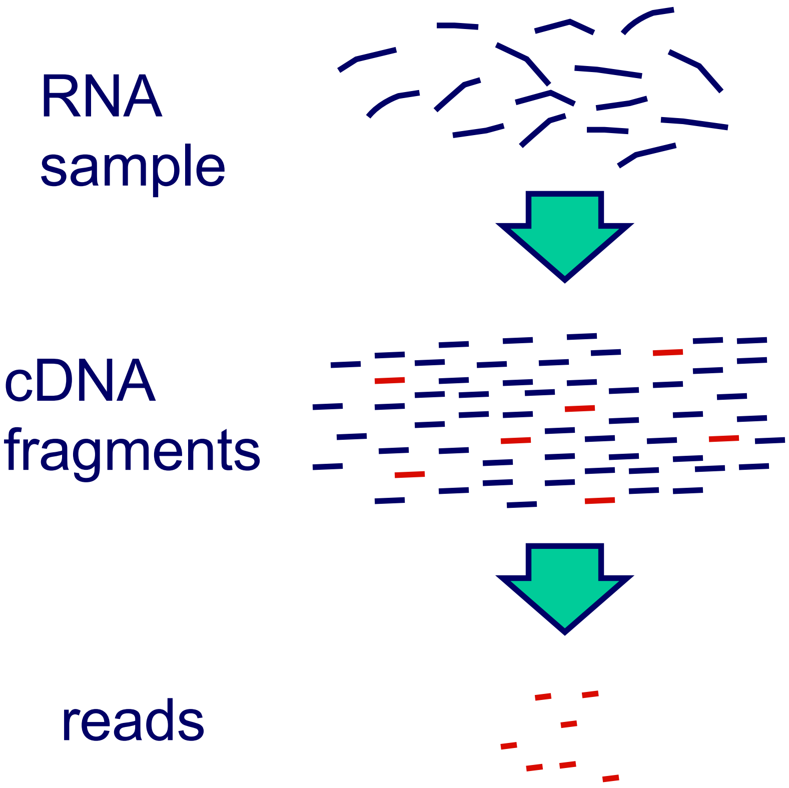
- **Assembly:**
  - Given: RNA-Seq reads (and possibly a genome sequence)
  - Do: Reconstruct full-length transcript sequences from the reads
- **Quantification (our focus):**
  - Given: RNA-Seq reads and transcript sequences
  - Do: Estimate the relative abundances of transcripts (“gene expression”)
- **Differential expression or additional downstream analyses:**
  - Given: RNA-Seq reads from two different samples and transcript sequences
  - Do: Predict which transcripts have different abundances between two samples

RNA-Seq is a *relative* abundance measurement technology

- RNA-Seq gives you reads from the ends of a random **sample** of fragments in your library

- Without additional data this only gives information about **relative** abundances

- Additional information, such as levels of “spike-in” transcripts, are needed for absolute measurements



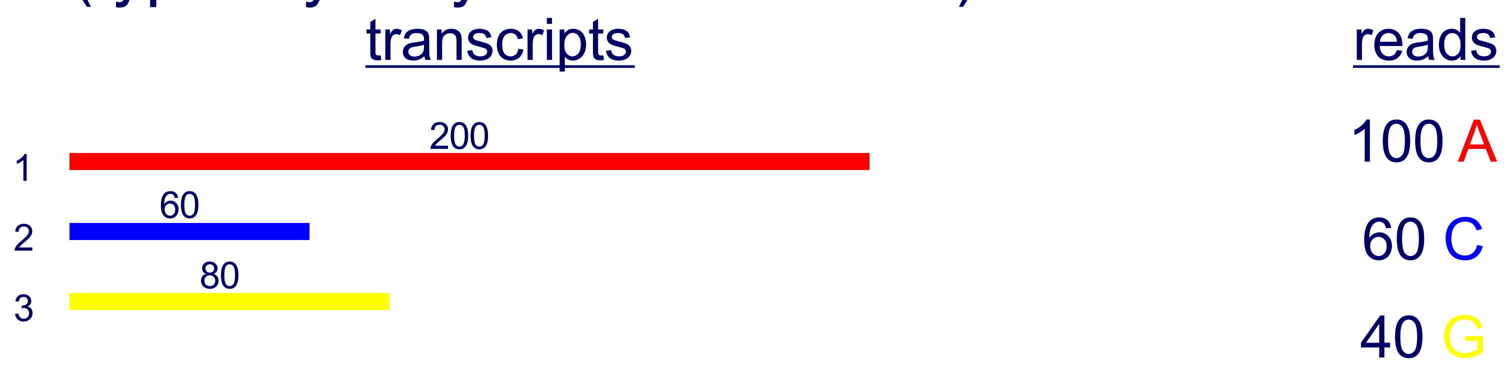
Issues with relative abundance measures

Gene	Sample 1 absolute abundance	Sample 1 relative abundance	Sample 2 absolute abundance	Sample 2 relative abundance
1	20	10%	20	5%
2	20	10%	20	5%
3	20	10%	20	5%
4	20	10%	20	5%
5	20	10%	20	5%
6	100	50%	300	75%

- Changes in absolute expression of high expressors is a major factor
- Normalization is required for comparing samples in these situations

The basics of quantification with RNA-Seq data

- For simplicity, suppose reads are of length **one** (typically they are > 35 bases)



transcripts

reads

1 200

2 60

3 80

100 A

60 C

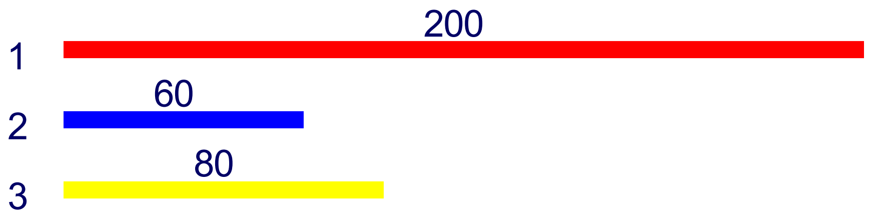
40 G

- What relative abundances would you estimate for these genes?
- Relative abundance is relative transcript levels in the cell, not proportion of observed reads

Length dependence

- Probability of a read coming from a transcript ∝ relative abundance × length

transcripts



reads

100 A

60 C

40 G

probability of read from transcript 1 = (transcript 1 reads) / (total reads)

transcript 1 relative abundance

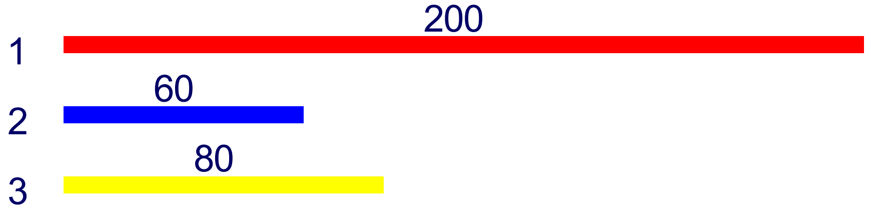
$$\hat{f}_1 \propto \frac{\frac{100}{200}}{200} = \frac{1}{400}$$

transcript 1 length

Length dependence

- Probability of a read coming from a transcript ∝ relative abundance × length

transcripts



reads

100 A

60 C

40 G

$$\hat{f}_1 \propto \frac{\frac{100}{200}}{200} = \frac{1}{400}$$

$$\hat{f}_2 \propto \frac{\frac{60}{200}}{60} = \frac{1}{200}$$

$$\hat{f}_3 \propto \frac{\frac{40}{200}}{80} = \frac{1}{400}$$

normalize

$$\hat{f}_1 = 0.25$$

$$\hat{f}_2 = 0.5$$

$$\hat{f}_3 = 0.25$$

The basics of quantification from RNA-Seq data

- Basic assumption:

$$\theta_i = P(\text{read from transcript } i) = Z^{-1}\tau_i \ell'_i$$

expression level
(relative abundance)

length

- Normalization factor is the mean length of expressed transcripts

$$Z = \sum_i \tau_i \ell'_i$$

The basics of quantification from RNA-Seq data

- Estimate the probability of reads being generated from a given transcript by counting the number of reads that align to that transcript

$$\hat{\theta}_i = \frac{c_i}{N}$$

# reads mapping to transcript i

total # of mappable reads

- Convert to expression levels by normalizing by transcript length

$$\hat{\tau}_i \propto \frac{\hat{\theta}_i}{\ell'_i}$$

The basics of quantification from RNA-Seq data

- Basic quantification algorithm
  - Align reads against a set of reference transcript sequences
  - Count the number of reads aligning to each transcript
  - Convert read counts into relative expression levels

Counts to expression levels

- RPKM - Reads Per Kilobase per Million mapped reads

$$\text{RPKM for gene i} = 10^9 \times \frac{c_i}{\ell'_i N}$$

- FPKM (fragments instead of reads, two reads per fragment, for paired end reads)
- TPM - Transcripts Per Million

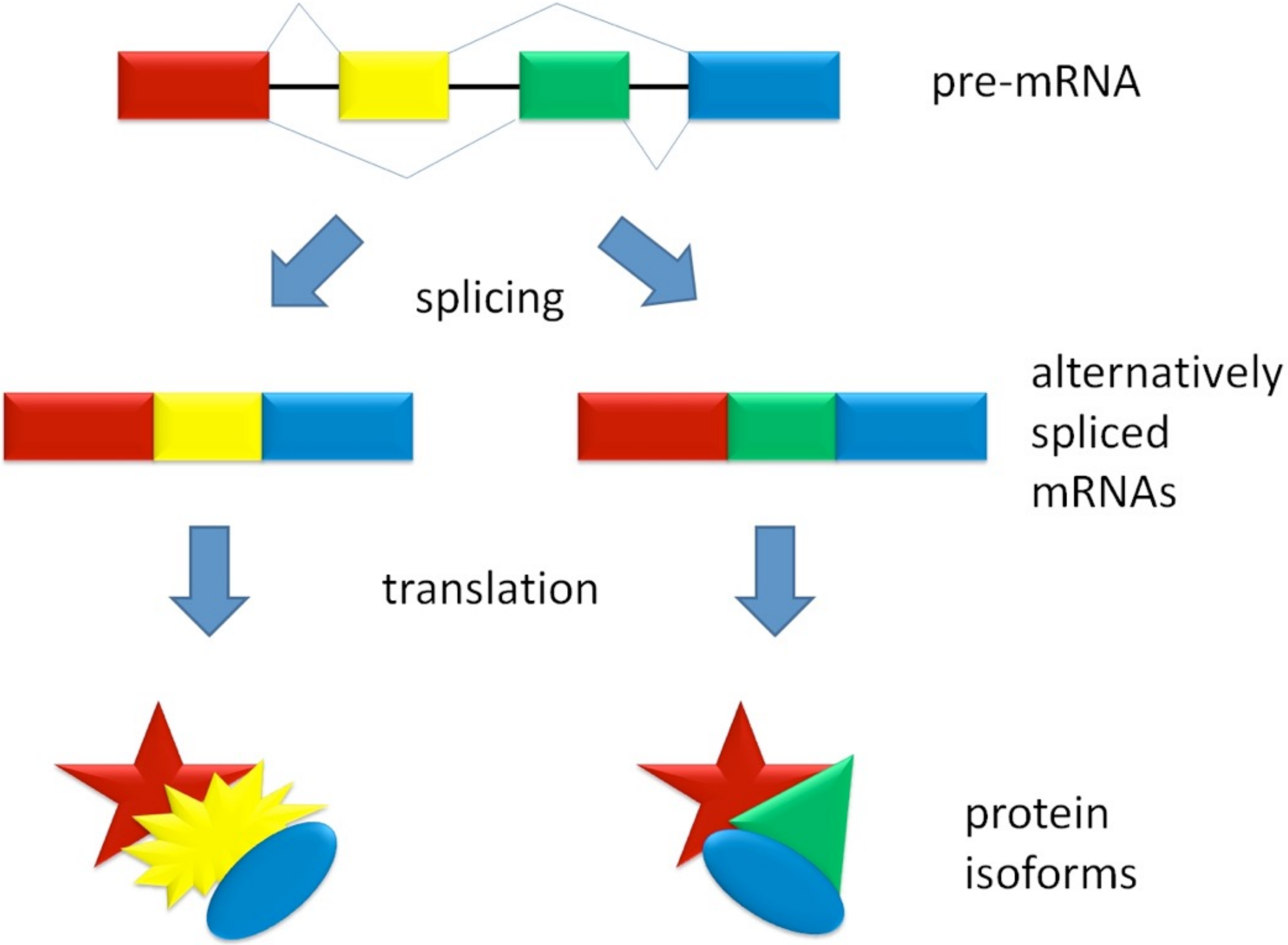
(estimate of) $\text{TPM for isoform i} = 10^6 \times Z \times \frac{c_i}{\ell'_i N}$

- Prefer TPM to RPKM because of normalization factor
  - TPM is a technology-independent measure (simply a fraction)

What if reads do not uniquely map to transcripts?

- The approach described assumes that every read can be uniquely aligned to a single transcript
- This is generally not the case
  - Some genes have similar sequences - gene families, repetitive sequences
  - Alternative splice forms of a gene share a significant fraction of sequence

Alternative splicing



pre-mRNA

splicing

alternatively spliced mRNAs

translation

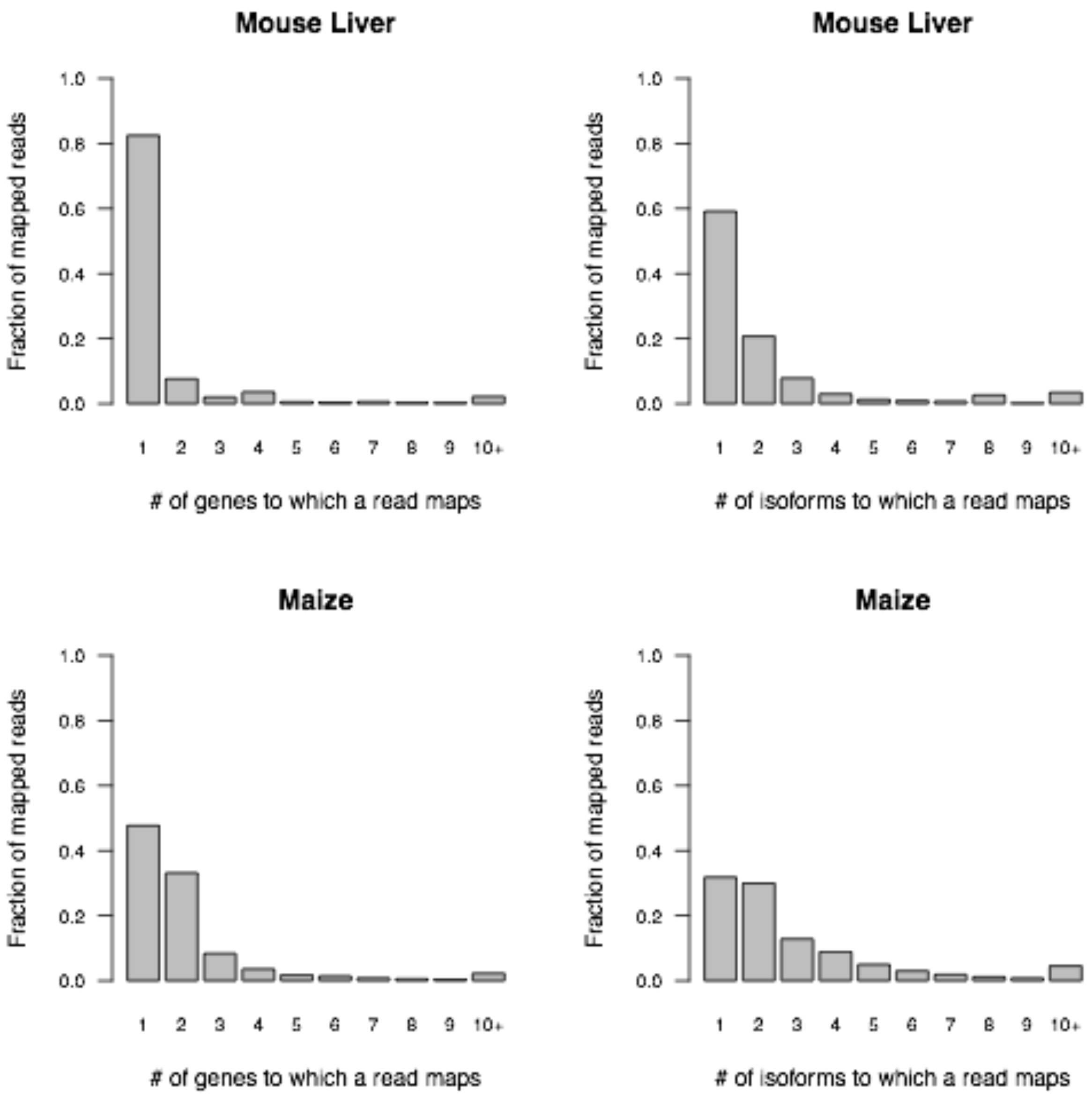
protein isoforms

Multi-mapping reads in RNA-Seq

Species	Read length	% multi-mapping reads
Mouse	25	17%
Mouse	75	10%
Maize	25	52%
Axolotl	76	23%
Human	50	23%

- Throwing away multi-mapping reads leads to
  - – Loss of information
  - – Potentially biased estimates of abundance

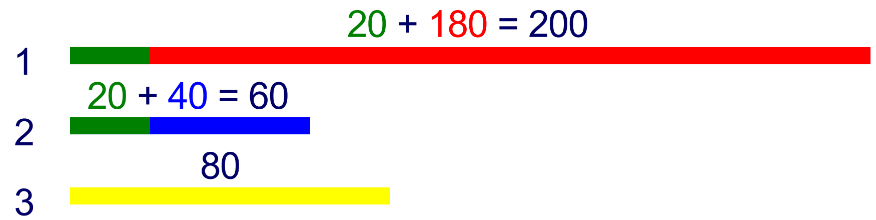
Distributions of alignment counts



What if reads do not uniquely map to transcripts?

- Multiread: a read that could have been derived from multiple transcripts

transcripts



reads

90 A

40 C

40 G

30 T

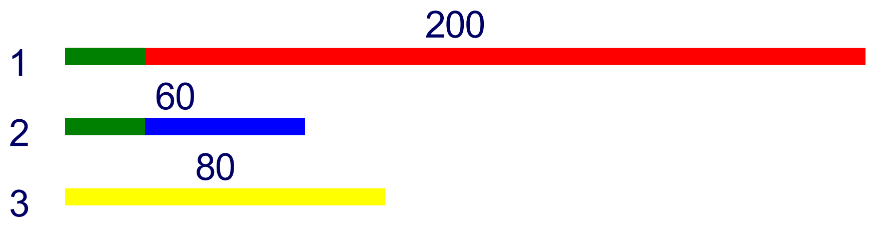
- How would you estimate the relative abundances for these transcripts?

Some options for handling multireads

- Discard multireads, estimate based on uniquely mapping reads only
- Discard multireads, but use "unique length" of each transcript in calculations
- "Rescue" multireads by allocating (fractions of) them to the transcripts
  - Three step algorithm
    1. Estimate abundances based on uniquely mapping reads only
    2. For each multiread, divide it between the transcripts to which it maps, proportionally to their abundances estimated in the first step
    3. Recompute abundances based on updated counts for each transcript

Rescue method example - Step 1

transcripts



reads

90 A

40 C

40 G

30 T

Step 1

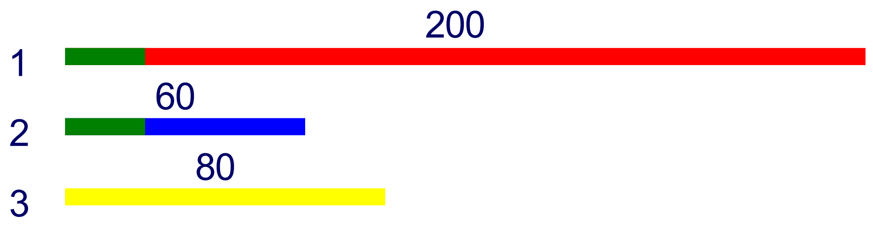
$$\hat{f}_1^{unique} = \frac{\frac{90}{200}}{\frac{90}{200} + \frac{40}{60} + \frac{40}{80}} = 0.278$$

$$\hat{f}_2^{unique} = 0.412$$

$$\hat{f}_3^{unique} = 0.309$$

Rescue method example - Step 2

transcripts



reads

90 A

40 C

40 G

30 T

Step 2

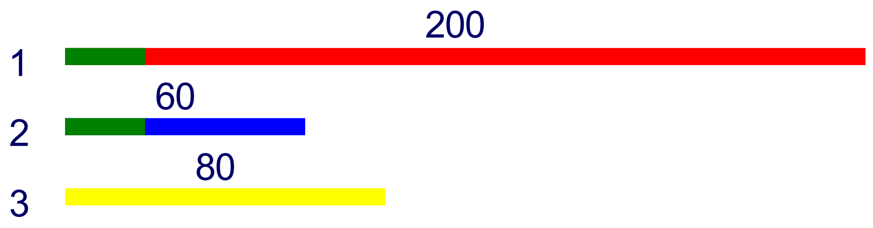
$$c_1^{rescue} = 90 + 30 \times \frac{0.278}{0.278 + 0.412} = 102.1$$

$$c_2^{rescue} = 40 + 30 \times \frac{0.412}{0.278 + 0.412} = 57.9$$

$$c_3^{rescue} = 40 + 0 = 40$$

Rescue method example - Step 3

transcripts



reads

90 A

40 C

40 G

30 T

Step 3

$$\hat{f}_1^{rescue} = \frac{\frac{102.1}{200}}{\frac{102.1}{200} + \frac{57.9}{60} + \frac{40}{80}} = 0.258$$

$$\hat{f}_2^{rescue} = \frac{\frac{57.9}{60}}{\frac{102.1}{200} + \frac{57.9}{60} + \frac{40}{80}} = 0.488$$

$$\hat{f}_3^{rescue} = \frac{\frac{40}{80}}{\frac{102.1}{200} + \frac{57.9}{60} + \frac{40}{80}} = 0.253$$

An observation about the rescue method

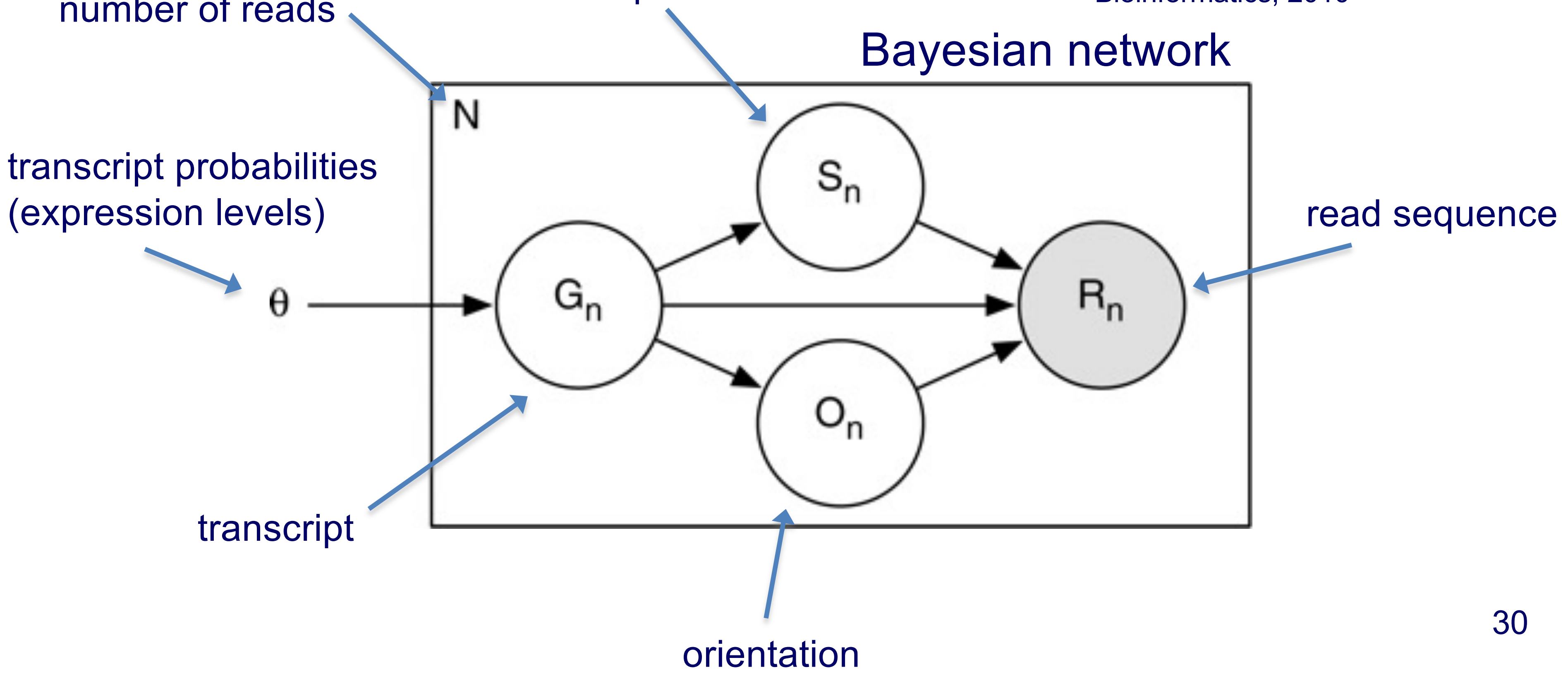
- Note that at the end of the rescue algorithm, we have an updated set of abundance estimates
- These new estimates could be used to reallocate the multireads
- And then we could update our abundance estimates once again
- And repeat!
- This is the intuition behind the statistical approach to this problem

RSEM (**R**NA-**S**eq by **E**xpectation-**M**aximization) - a generative probabilistic model

- Simplified view of the model (plate notation)
  - Grey – observed variable
  - White – latent (unobserved) variables

"RNA-Seq gene expression estimation with read mapping uncertainty"
Li, B., Ruotti, V., Stewart, R., Thomson, J., Dewey, C.
Bioinformatics, 2010

Bayesian network



number of reads

start position

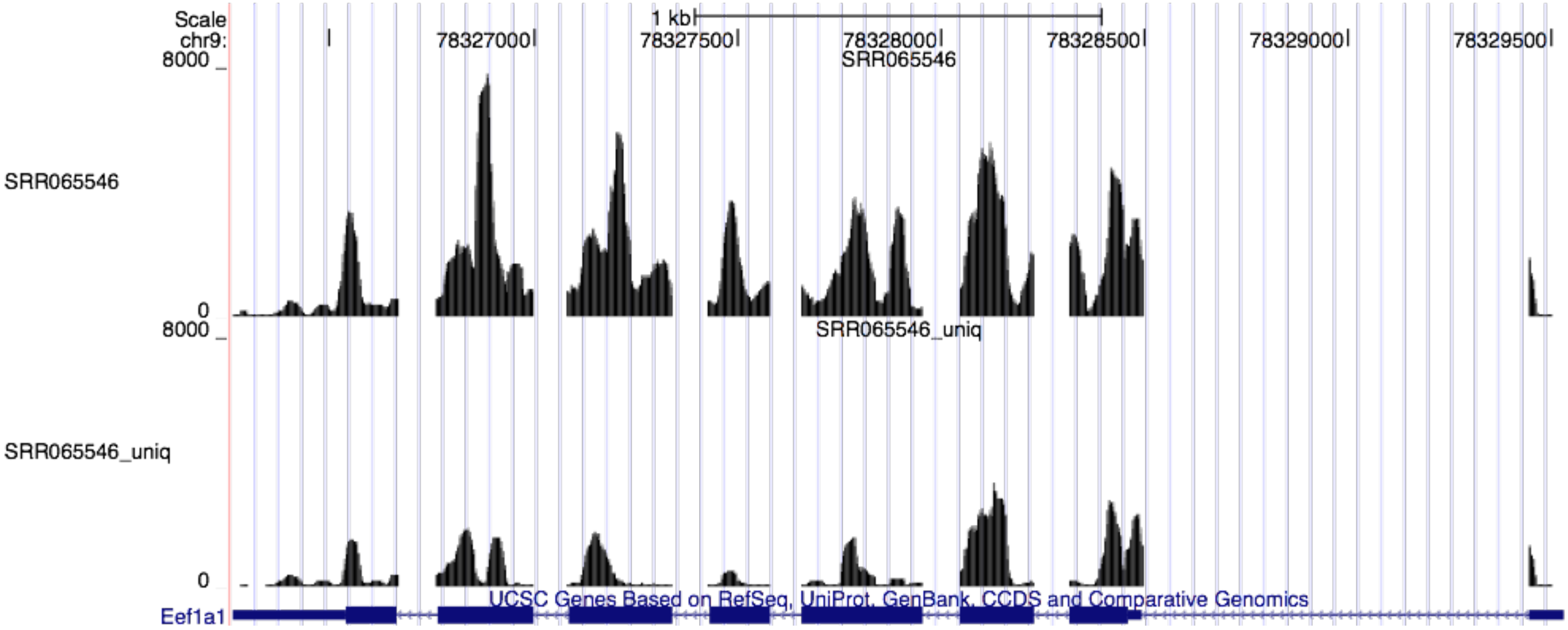
transcript probabilities (expression levels)

read sequence

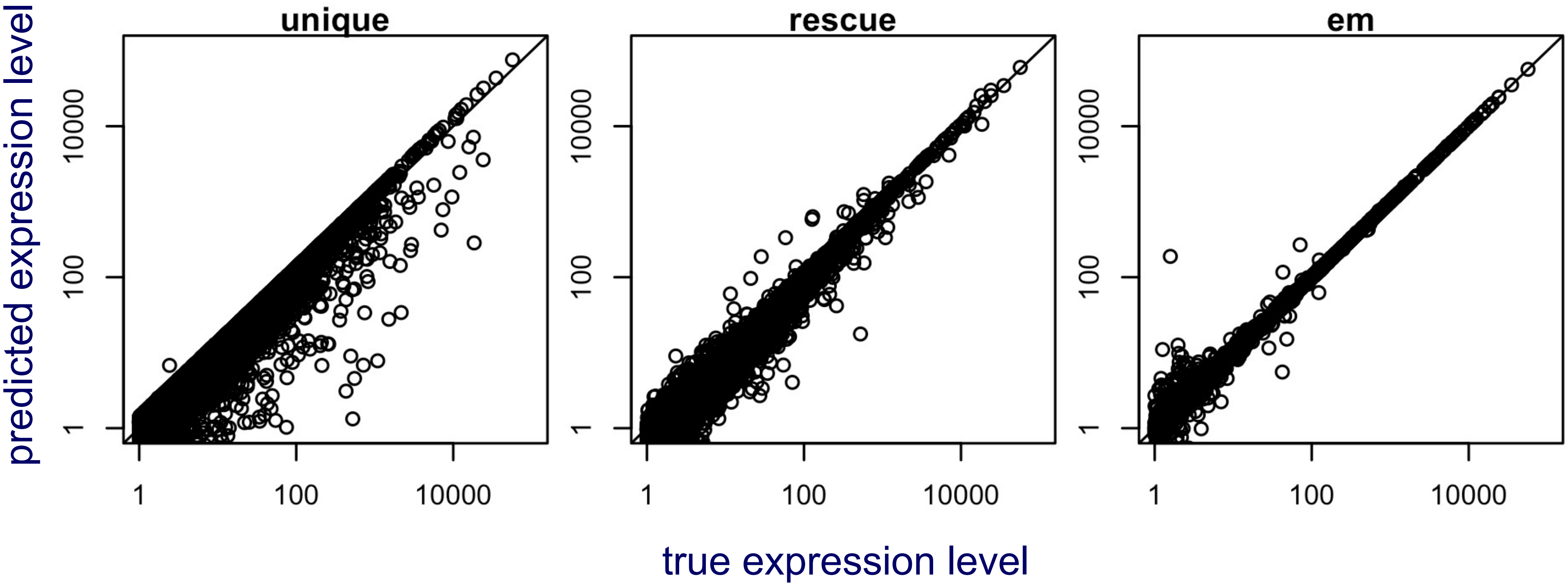
transcript

orientation

Expected read count visualization



Improved accuracy over unique and rescue



Mouse gene-level expression estimation

RNA-Seq summary

- RNA-Seq is the preferred technology for transcriptome analysis in most settings
- The major challenge in analyzing RNA-Seq data: the reads are much shorter than the transcripts from which they are derived
- Tasks with RNA-Seq data thus require handling hidden information: which gene/isoform gave rise to a given read
- The Expectation-Maximization algorithm is extremely powerful in these situations, e.g., RSEM

Recent developments in RNA-Seq

- Long read sequences: PacBio and Oxford Nanopore

- Single-cell RNA-Seq: review
  - Observe heterogeneity of cell populations
  - Model technical artifacts (e.g. artificial 0 counts)
  - Detect sub-populations
  - Predict pseudotime through dynamic processes
  - Detect gene-gene and cell-cell relationships

- Alignment-free quantification:
  - Kallisto
  - Salmon

Public sources of RNA-Seq data

- Gene Expression Omnibus (GEO): http://www.ncbi.nlm.nih.gov/geo/
  - Both microarray and sequencing data
- Sequence Read Archive (SRA): http://www.ncbi.nlm.nih.gov/sra
  - All sequencing data (not necessarily RNA-Seq)
- ArrayExpress: https://www.ebi.ac.uk/arrayexpress/
  - European version of GEO
- Homogenized data: MetaSRA, Toil, recount2, ARCHS^4

Interpolated Markov Models for Gene Finding

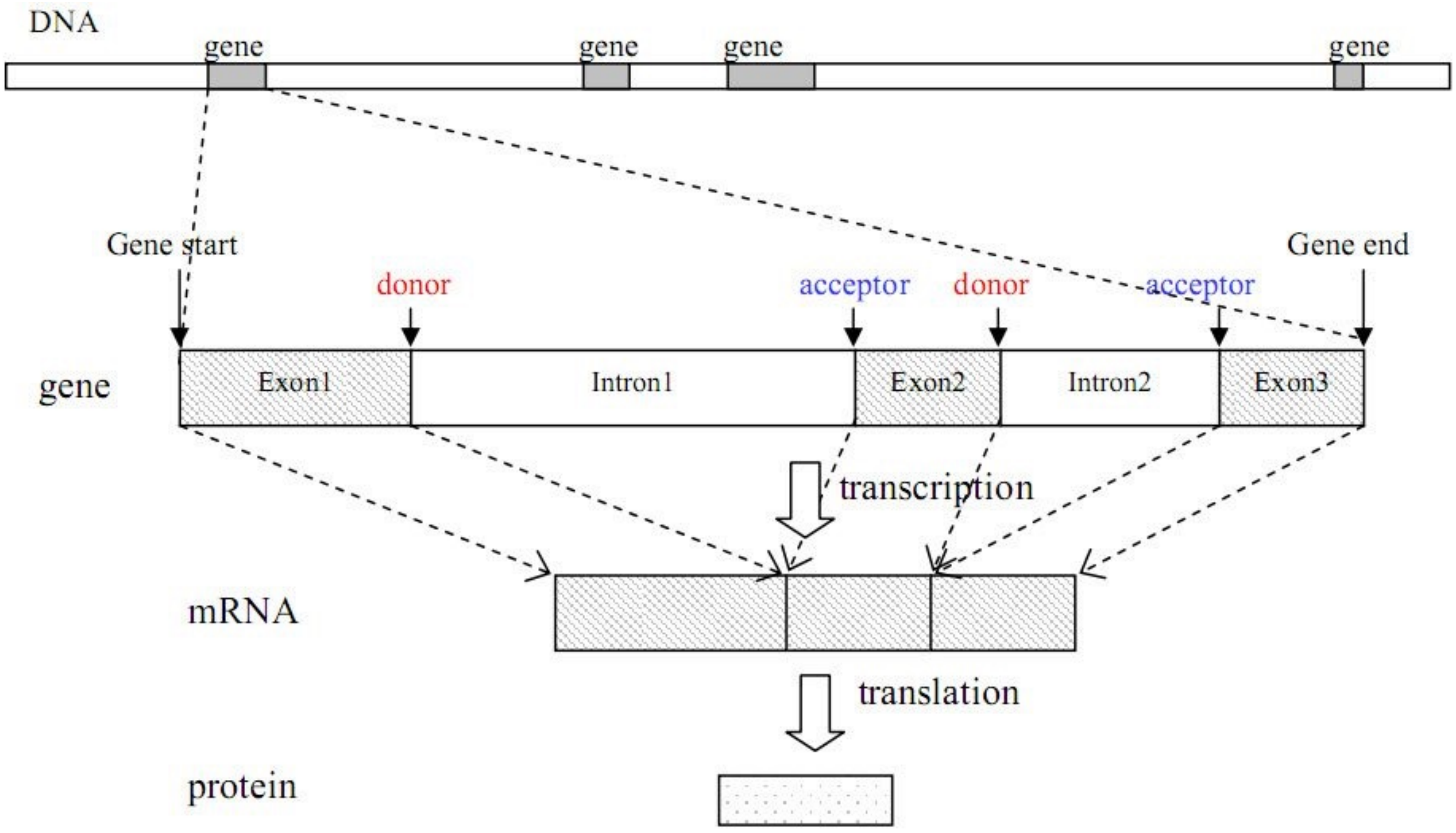
Key concepts

- the gene-finding task
- the trade-off between potential predictive value and parameter uncertainty in choosing the order of a Markov model
- interpolated Markov models

The Gene Finding Task

Given: an uncharacterized DNA sequence

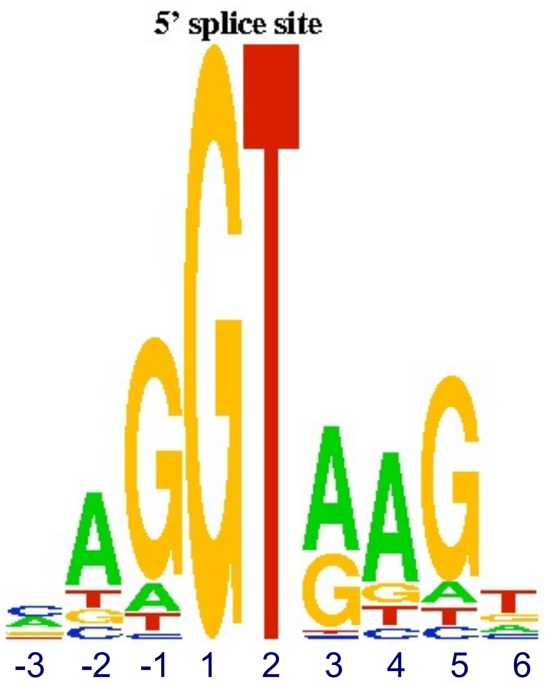
Do: locate the genes in the sequence, including the coordinates of individual *exons* and *introns*

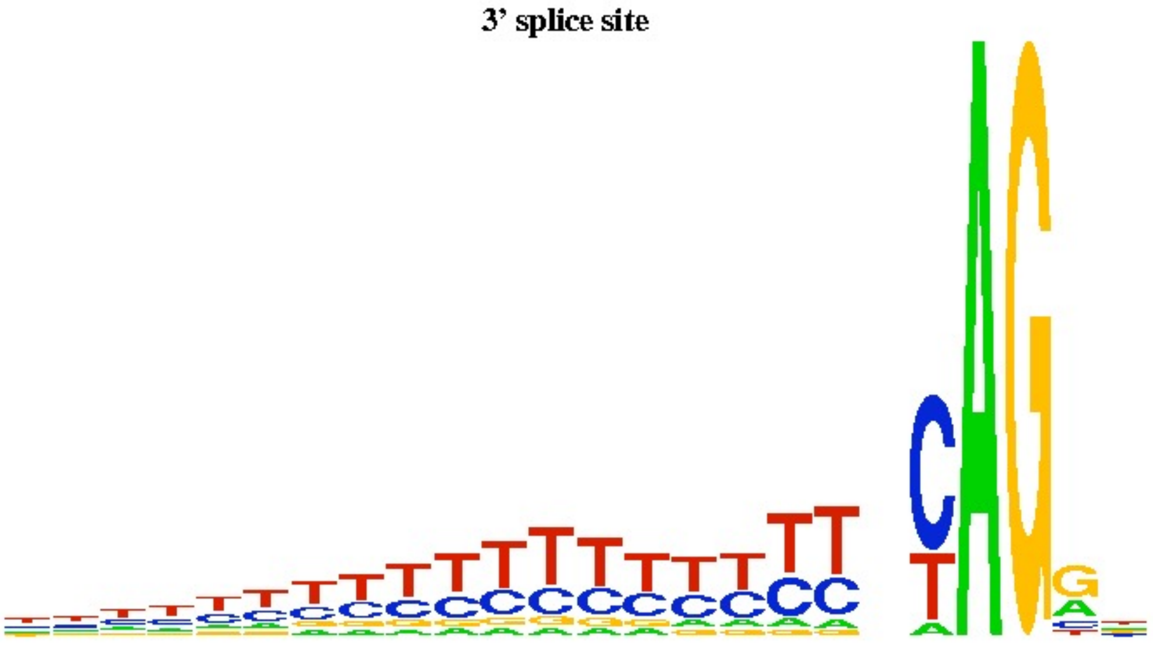


Splice Signals Example

donor sites

acceptor sites





exon

Figures from Yi Xing

exon

- There are significant dependencies among non-adjacent positions in donor splice signals
- Informative for inferring hidden state of HMM

Sources of Evidence for Gene Finding

- **Signals**: the sequence *signals* (e.g. splice junctions) involved in gene expression (e.g., RNA-seq reads)

- **Content**: statistical properties that distinguish protein-coding DNA from non-coding DNA (focus in this lecture)

- **Conservation**: signal and content properties that are conserved across related sequences (e.g. orthologous regions of the mouse and human genome)

Gene Finding: Search by Content

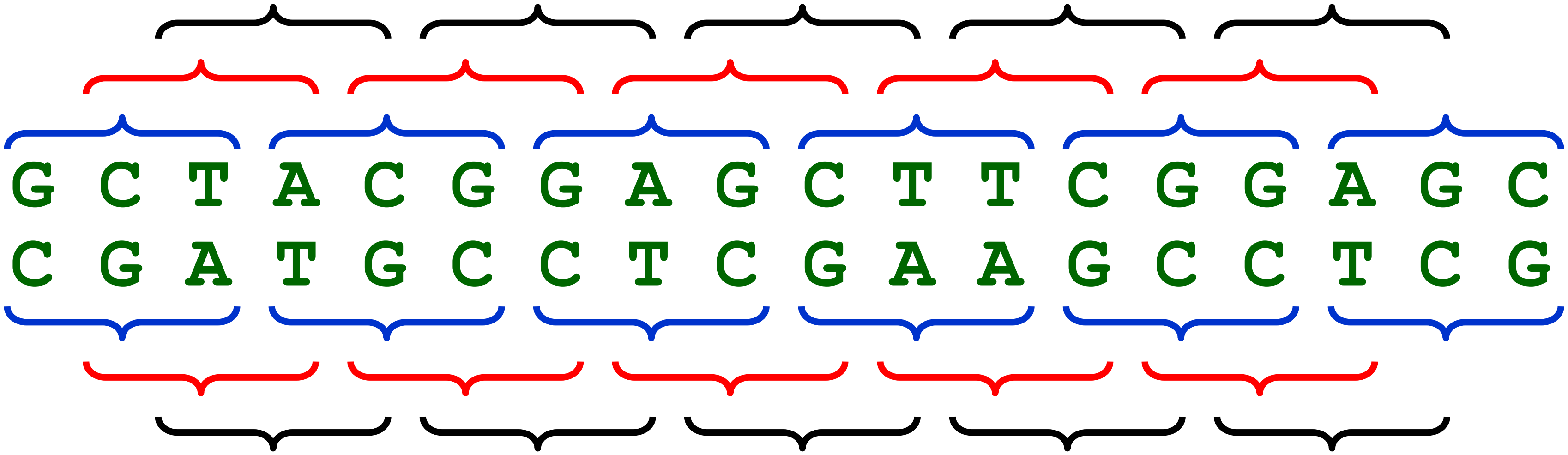
- Encoding a protein affects the statistical properties of a DNA sequence
  - some amino acids are used more frequently than others (Leu more prevalent than Trp)
  - different numbers of codons for different amino acids (Leu has 6, Trp has 1)
  - for a given amino acid, usually one codon is used more frequently than others
    - this is termed *codon preference*
    - these preferences vary by species

Codon Preference in E. Coli

AA	codon	/1000
Gly	GGG	1.89
Gly	GGA	0.44
Gly	GGU	52.99
Gly	GGC	34.55
Glu	GAG	15.68
Glu	GAA	57.20
Asp	GAU	21.63
Asp	GAC	43.26

Reading Frames

- A given sequence may encode a protein in any of the six reading frames (three on each strand)

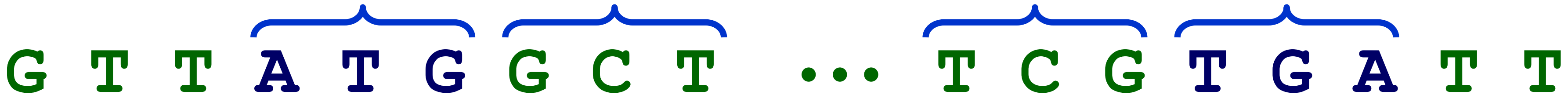


G C T A C G G A G C T T C G G A G C

C G A T G C C T C G A A G C C T C G

Open Reading Frames (ORFs)

- An ORF is a sequence that
  - starts with a potential start codon (e.g., ATG)
  - ends with a potential stop codon, *in the same reading frame* (e.g., TAG, TAA, TGA)
  - doesn't contain another stop codon in-frame
  - and is sufficiently long (say > 100 bases)

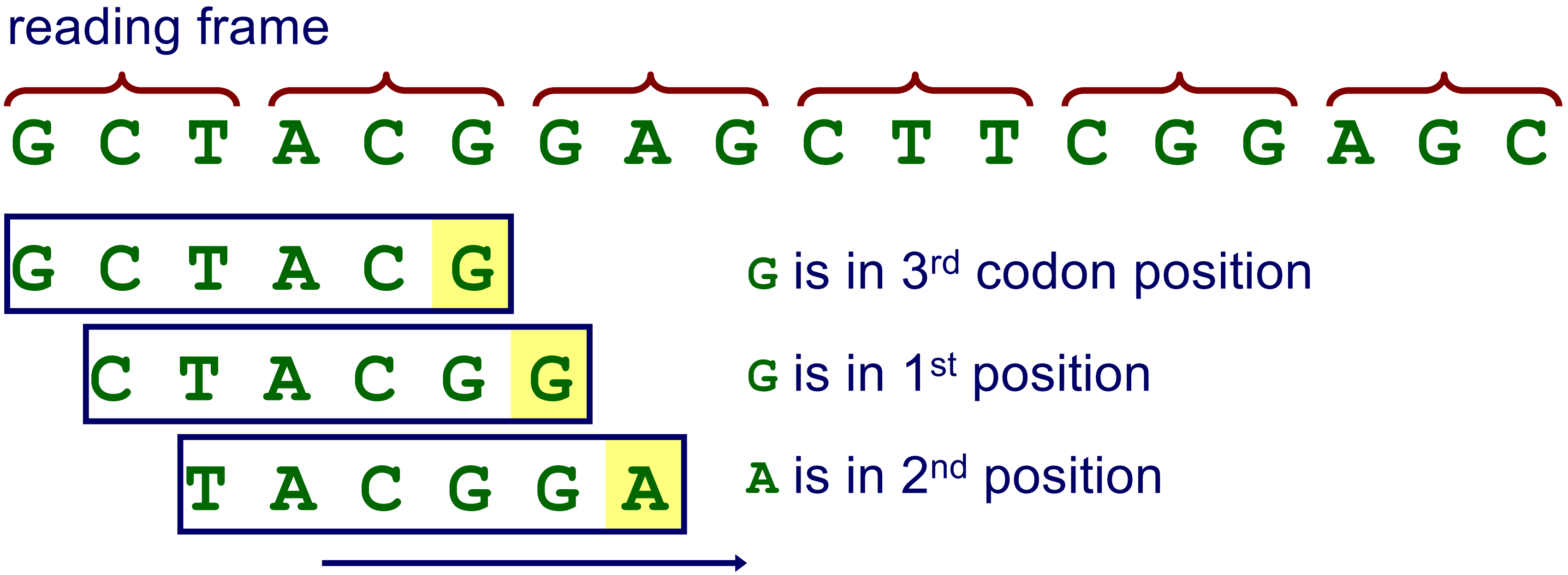


G T T A T G G C T … T C G T G A T T

- An ORF meets the minimal requirements to be a protein-coding gene in an organism without introns
- NHGRI ORF

Markov Models & Reading Frames

- Consider modeling a given coding sequence
- For each “word” we evaluate, we’ll want to consider its position with respect to the reading frame we’re assuming



reading frame

G C T A C G G A G C T T C G G A G C

G C T A C G

C T A C G G

T A C G G A

G is in 3rd codon position

G is in 1st position

A is in 2nd position

- Can do this using an inhomogeneous model

Inhomogeneous Markov Model

- **Homogenous Markov model**: transition probability matrix does not change over time or position

- **Inhomogenous Markov model**: transition probability matrix depends on the time or position

Higher Order Markov Models

- Higher order models remember more "history"
  - n-order $P(x_i \mid x_{i-1}, x_{i-2}, \ldots, x_1) = P(x_i \mid x_{i-1}, \ldots, x_{i-n})$
- Additional history can have predictive value
- Example:
  - predict the next word in this sentence fragment "…you__" (are, give, passed, say, see, too, …?)
  - now predict it given more history

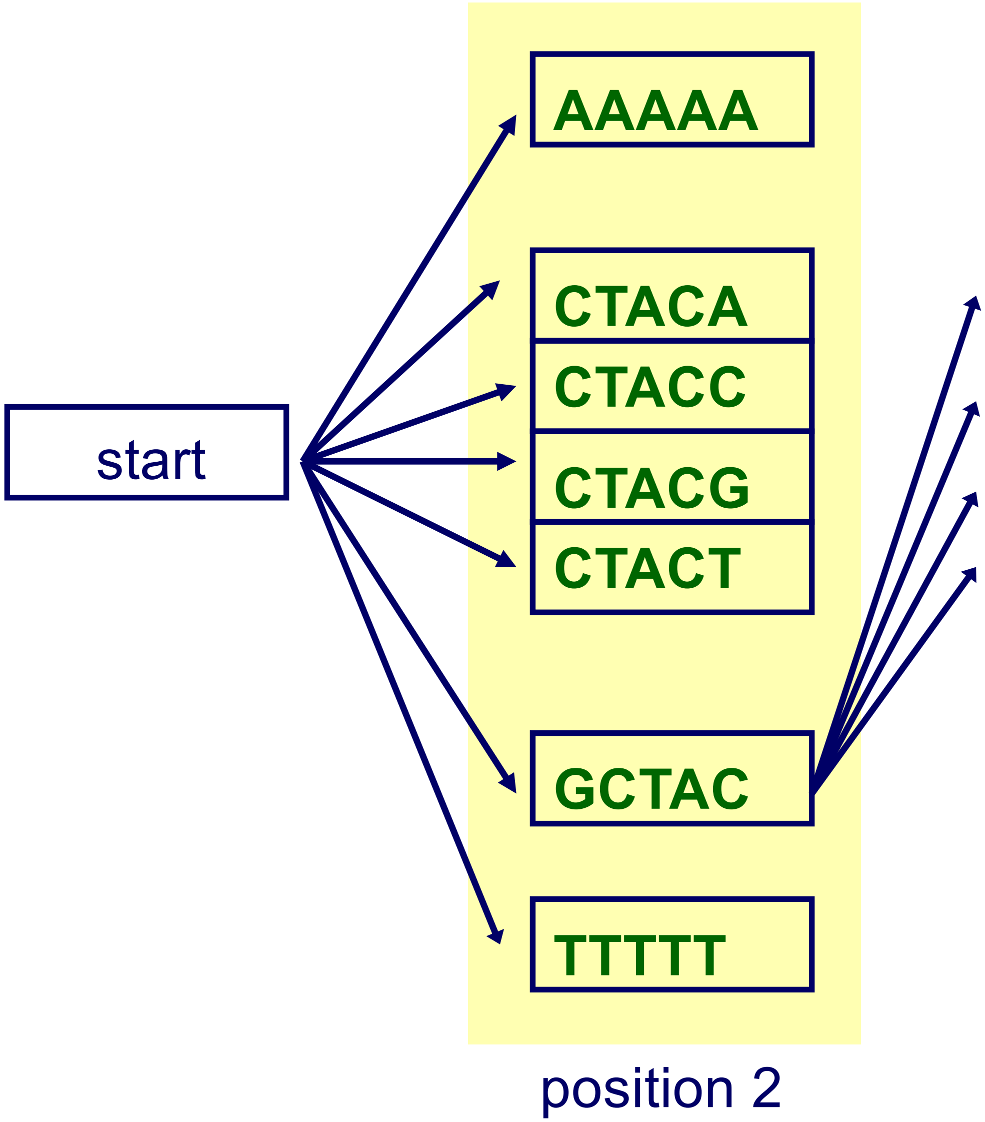
"…can you___"

"…say can you___"

"…oh say can you___"

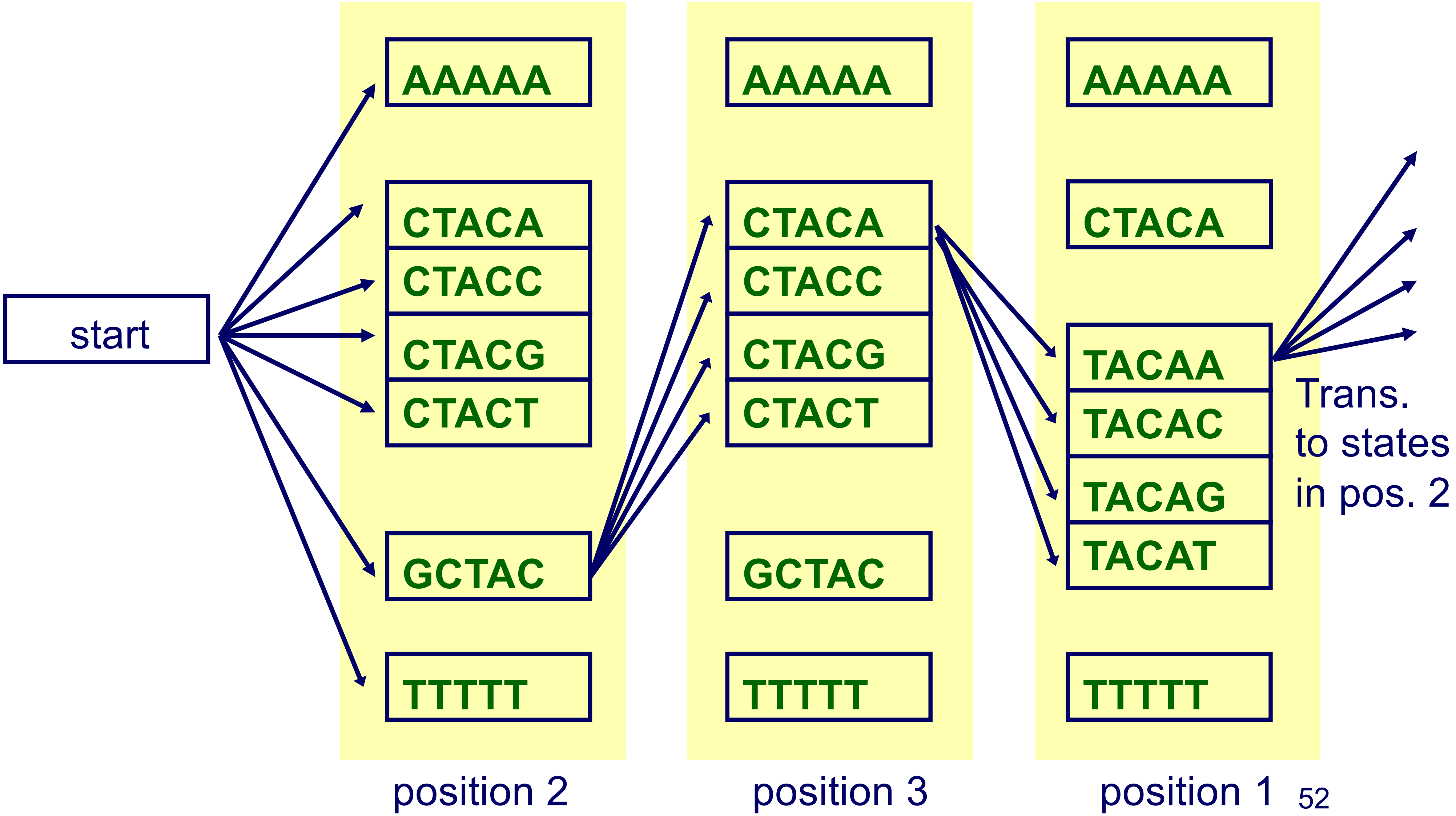
YouTube

A Fifth Order Inhomogeneous Markov Model



$$P(x_i \mid x_{i-5}, \ldots, x_{i-1}, position)$$

A Fifth Order Inhomogeneous Markov Model



Selecting the Order of a Markov Model

- But the number of parameters we need to estimate grows exponentially with the order
  - for modeling DNA we need $O(4^{n+1})$ parameters for an *n*th order model

- The higher the order, the less reliable we can expect our parameter estimates to be
- Suppose we have 100k bases of sequence to estimate parameters of a model
  - for a 2nd order homogeneous Markov chain, we'd see each history 6250 times on average
  - for an 8th order chain, we'd see each history ~ 1.5 times on average

Interpolated Markov Models

- The IMM idea: manage this trade-off by interpolating among models of various orders
- *Simple* linear interpolation:

$$
\begin{aligned}
P_{\mathrm{IMM}}(x_i \mid x_{i-n},\ldots,x_{i-1}) = {} & \lambda_0 P(x_i) \\
& + \lambda_1 P(x_i \mid x_{i-1}) \\
& \cdots \\
& + \lambda_n P(x_i \mid x_{i-n},\ldots,x_{i-1})
\end{aligned}
$$

- where $\sum_i \lambda_i = 1$

Interpolated Markov Models

- We can make the weights depend on the history
  - for a given order, we may have significantly more data to estimate some words than others
- *General* linear interpolation

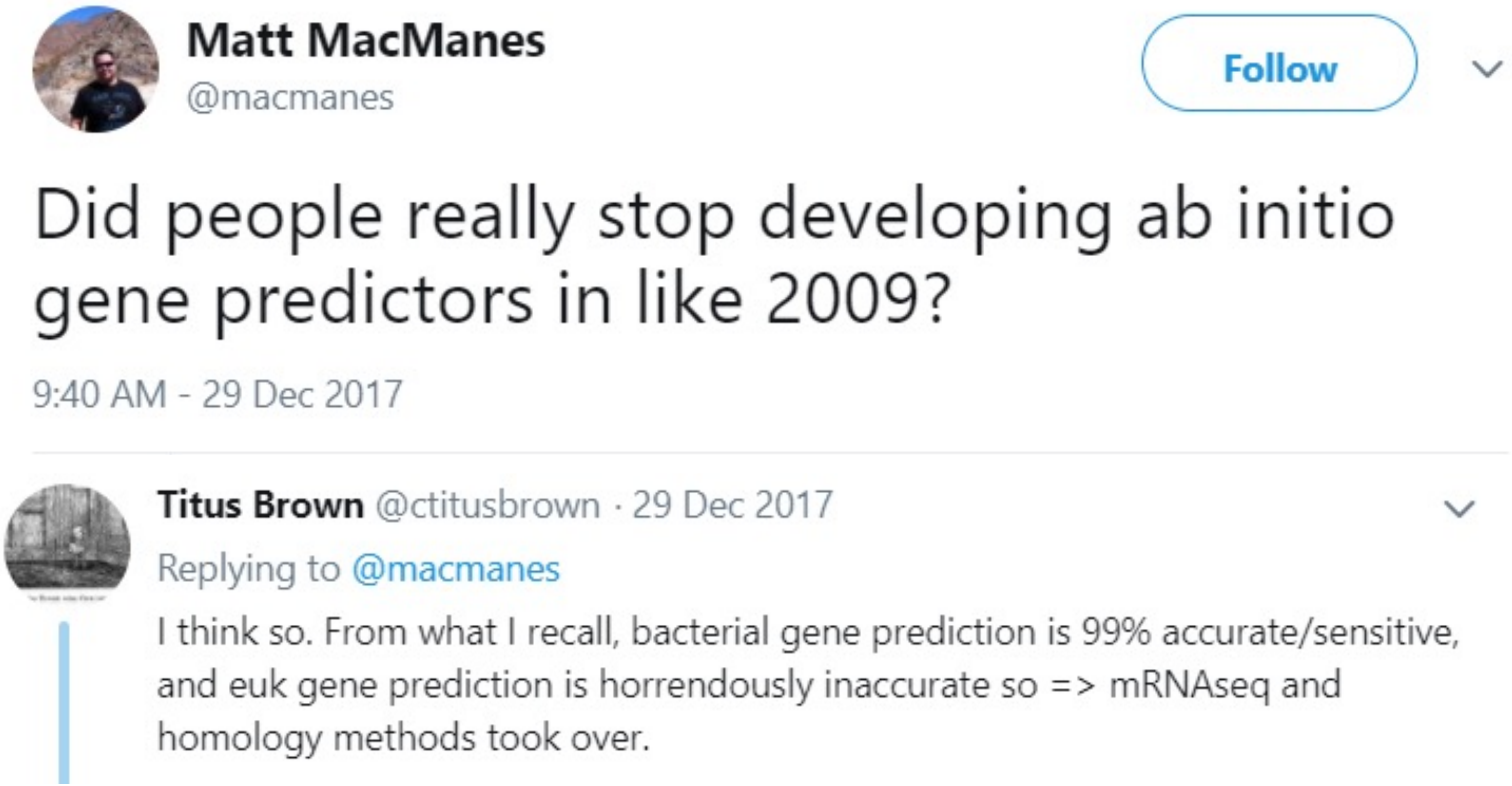
$$\begin{aligned} P_{\text{IMM}}(x_i \mid x_{i-n},\ldots,x_{i-1}) = {} & \lambda_0 P(x_i) \\ & + \lambda_1(x_{i-1}) P(x_i \mid x_{i-1}) \\ & \ldots \\ & + \lambda_n(x_{i-n},\ldots,x_{i-1}) P(x_i \mid x_{i-n},\ldots,x_{i-1}) \end{aligned}$$

λ is a function of the given history

The GLIMMER System

[Salzberg et al., Nucleic Acids Research, 1998]

- System for identifying genes in bacterial genomes
- Uses 8th order, inhomogeneous, interpolated Markov models



Matt MacManes
@macmanes

Did people really stop developing ab initio gene predictors in like 2009?

9:40 AM - 29 Dec 2017

Titus Brown @ctitusbrown · 29 Dec 2017

Replying to @macmanes

I think so. From what I recall, bacterial gene prediction is 99% accurate/sensitive, and euk gene prediction is horrendously inaccurate so => mRNAseq and homology methods took over.

IMMs in GLIMMER

- How does GLIMMER determine the λ values?
- First, let’s express the IMM probability calculation recursively

$$
\begin{aligned}
&P_{\mathrm{IMM,n}}(x_i \mid x_{i-n},\ldots,x_{i-1}) = \\
&\quad \lambda_{\mathrm{n}}(x_{i-n},\ldots,x_{i-1})P(x_i \mid x_{i-n},\ldots,x_{i-1}) + \\
&\quad [1-\lambda_n(x_{i-n},\ldots,x_{i-1})]P_{\mathrm{IMM,n\text{-}1}}(x_i \mid x_{i-n+1},\ldots,x_{i-1})
\end{aligned}
$$

- Let $c(x_{i-n},\ldots,x_{i-1})$ be the number of times we see the history $x_{i-n},\ldots,x_{i-1}$ in our training set

$$\lambda_n(x_{i-n},\ldots,x_{i-1}) = 1 \quad \text{if} \quad c(x_{i-n},\ldots,x_{i-1}) > 400$$

IMMs in GLIMMER

- If we haven’t seen $x_{i-n},\ldots,x_{i-1}$ more than 400 times, then compare the counts for the following:

*n*th order history + base	(*n-1*)th order history + base
$x_{i-n},\ldots,x_{i-1},a$	$x_{i-n+1},\ldots,x_{i-1},a$
$x_{i-n},\ldots,x_{i-1},c$	$x_{i-n+1},\ldots,x_{i-1},c$
$x_{i-n},\ldots,x_{i-1},g$	$x_{i-n+1},\ldots,x_{i-1},g$
$x_{i-n},\ldots,x_{i-1},t$	$x_{i-n+1},\ldots,x_{i-1},t$

- Use a statistical test to assess whether the distributions of x_i depend on the order

IMMs in GLIMMER

*n*th order history + base	(*n-1*)th order history + base
$x_{i-n},\ldots,x_{i-1},a$	$x_{i-n+1},\ldots,x_{i-1},a$
$x_{i-n},\ldots,x_{i-1},c$	$x_{i-n+1},\ldots,x_{i-1},c$
$x_{i-n},\ldots,x_{i-1},g$	$x_{i-n+1},\ldots,x_{i-1},g$
$x_{i-n},\ldots,x_{i-1},t$	$x_{i-n+1},\ldots,x_{i-1},t$

- Null hypothesis in χ^2 test: x_i distribution is independent of order
- Define $d = 1 - pvalue$
- If d is small we don't need the higher order history

IMMs in GLIMMER

- Putting it all together

$$\lambda_n(x_{i-n},\ldots,x_{i-1}) = \begin{cases} 1 & \text{if } c(x_{i-n},\ldots,x_{i-1}) > 400 \\ d \times \dfrac{c(x_{i-n},\ldots,x_{i-1})}{400} & \text{else if } d \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

where $d \in (0,1)$

- why 400?
  - “gives ~95% confidence that the sample probabilities are within ±0.05 of the true probabilities from which the sample was taken”

IMM Example

- Suppose we have the following counts from our training set

ACGA	25	**CGA**	100	**GA**	175
ACGC	40	**CGC**	90	**GC**	140
ACGG	15	**CGG**	35	**GG**	65
ACGT	20	**CGT**	75	**GT**	120
	100		300		500

χ^2 test: $d = 0.857$ χ^2 test: $d = 0.140$

$$\lambda_3(\mathbf{ACG}) = 0.857 \times 100/400 = 0.214$$

$$\lambda_2(\mathbf{CG}) = 0 \quad (d < 0.5,\ c(\mathbf{CG}) < 400)$$

$$\lambda_1(\mathbf{G}) = 1 \quad (c(\mathbf{G}) > 400)$$

IMM Example (Continued)

- Now suppose we want to calculate $P_{\mathrm{IMM},3}(T \mid ACG)$

$$
\begin{aligned}
P_{\mathrm{IMM},1}(T \mid G) &= \lambda_1(G)P(T \mid G) + \big(1-\lambda_1(G)\big)P_{\mathrm{IMM},0}(T) \\
&= P(T \mid G)
\end{aligned}
$$

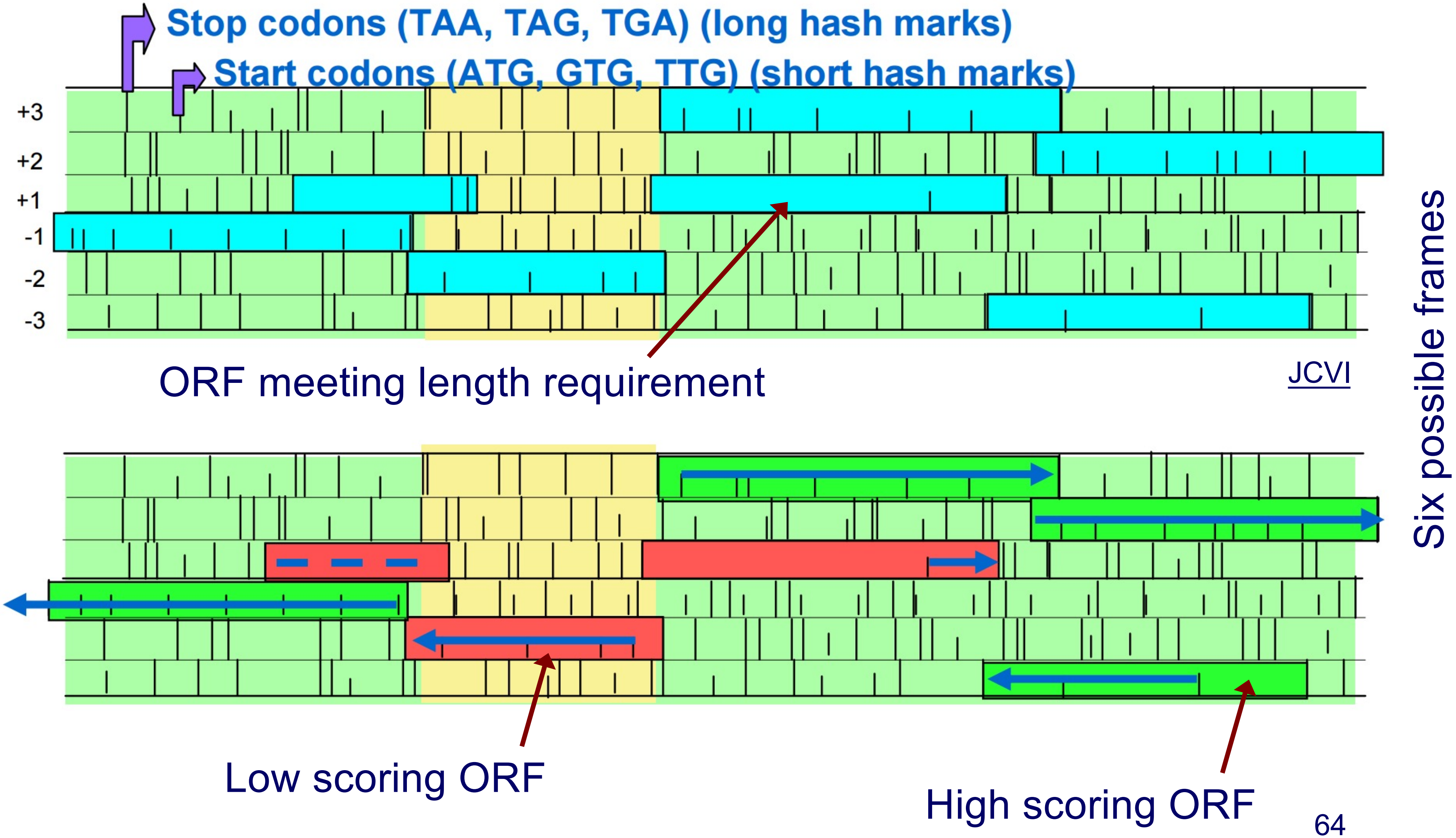
$$
\begin{aligned}
P_{\mathrm{IMM},2}(T \mid CG) &= \lambda_2(CG)P(T \mid CG) + \big(1-\lambda_2(CG)\big)P_{\mathrm{IMM},1}(T \mid G) \\
&= P(T \mid G)
\end{aligned}
$$

$$
\begin{aligned}
P_{\mathrm{IMM},3}(T \mid ACG) &= \lambda_3(ACG)P(T \mid ACG) + \big(1-\lambda_3(ACG)\big)P_{\mathrm{IMM},2}(T \mid CG) \\
&= 0.214 \times P(T \mid ACG) + (1-0.214) \times P(T \mid G) \\
&= 0.214 \times 0.2 + (1-0.214) \times 0.24
\end{aligned}
$$

Gene Recognition in GLIMMER

- Essentially ORF classification
  - Train and estimate IMMs
- For each ORF
  - calculate the probability of the ORF sequence in each of the 6 possible reading frames
  - if the highest scoring frame corresponds to the reading frame of the ORF, mark the ORF as a gene
- For overlapping ORFs that look like genes
  - score overlapping region separately
  - predict only one of the ORFs as a gene

Gene Recognition in GLIMMER



Stop codons (TAA, TAG, TGA) (long hash marks)

Start codons (ATG, GTG, TTG) (short hash marks)

ORF meeting length requirement

JCVI

Six possible frames

Low scoring ORF

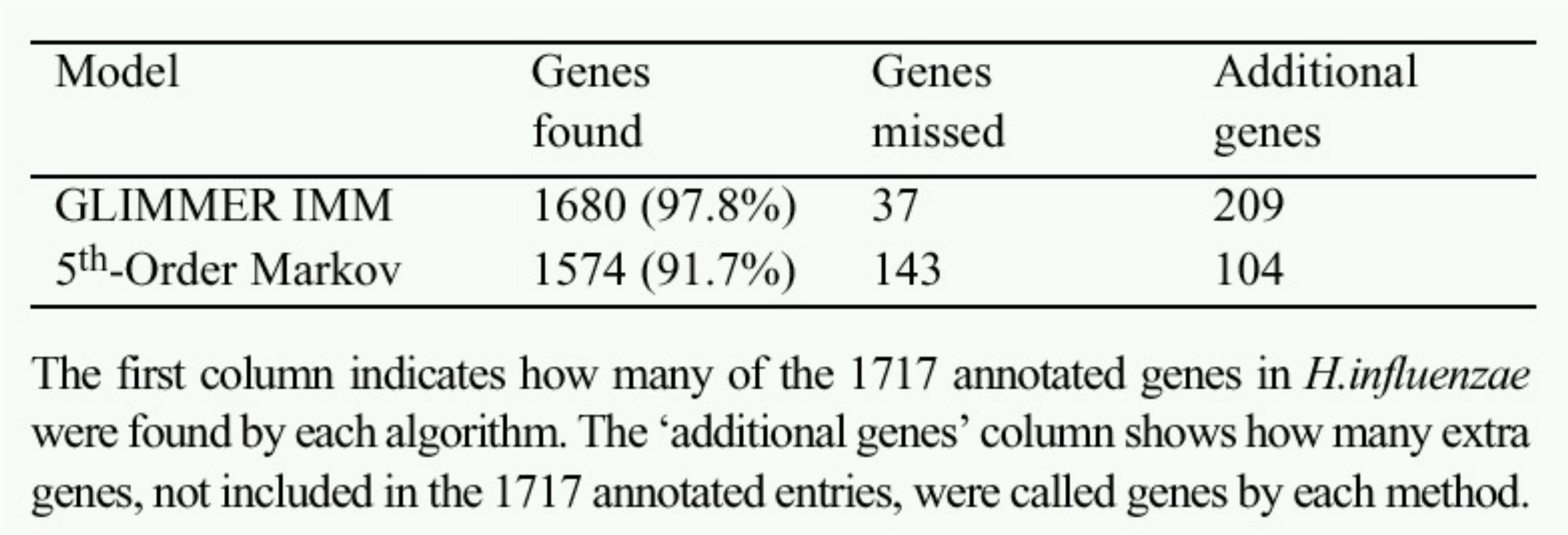
High scoring ORF

GLIMMER Experiment

- 8th order IMM vs. 5th order Markov model
- Trained on 1168 genes (ORFs really)
- Tested on 1717 annotated (more or less known) genes

GLIMMER Results

TP FN FP & TP?



Model	Genes found	Genes missed	Additional genes
GLIMMER IMM	1680 (97.8%)	37	209
5th-Order Markov	1574 (91.7%)	143	104

The first column indicates how many of the 1717 annotated genes in *H.influenzae* were found by each algorithm. The ‘additional genes’ column shows how many extra genes, not included in the 1717 annotated entries, were called genes by each method.

- GLIMMER has greater sensitivity than the baseline
- It’s not clear whether its precision/specificity is better