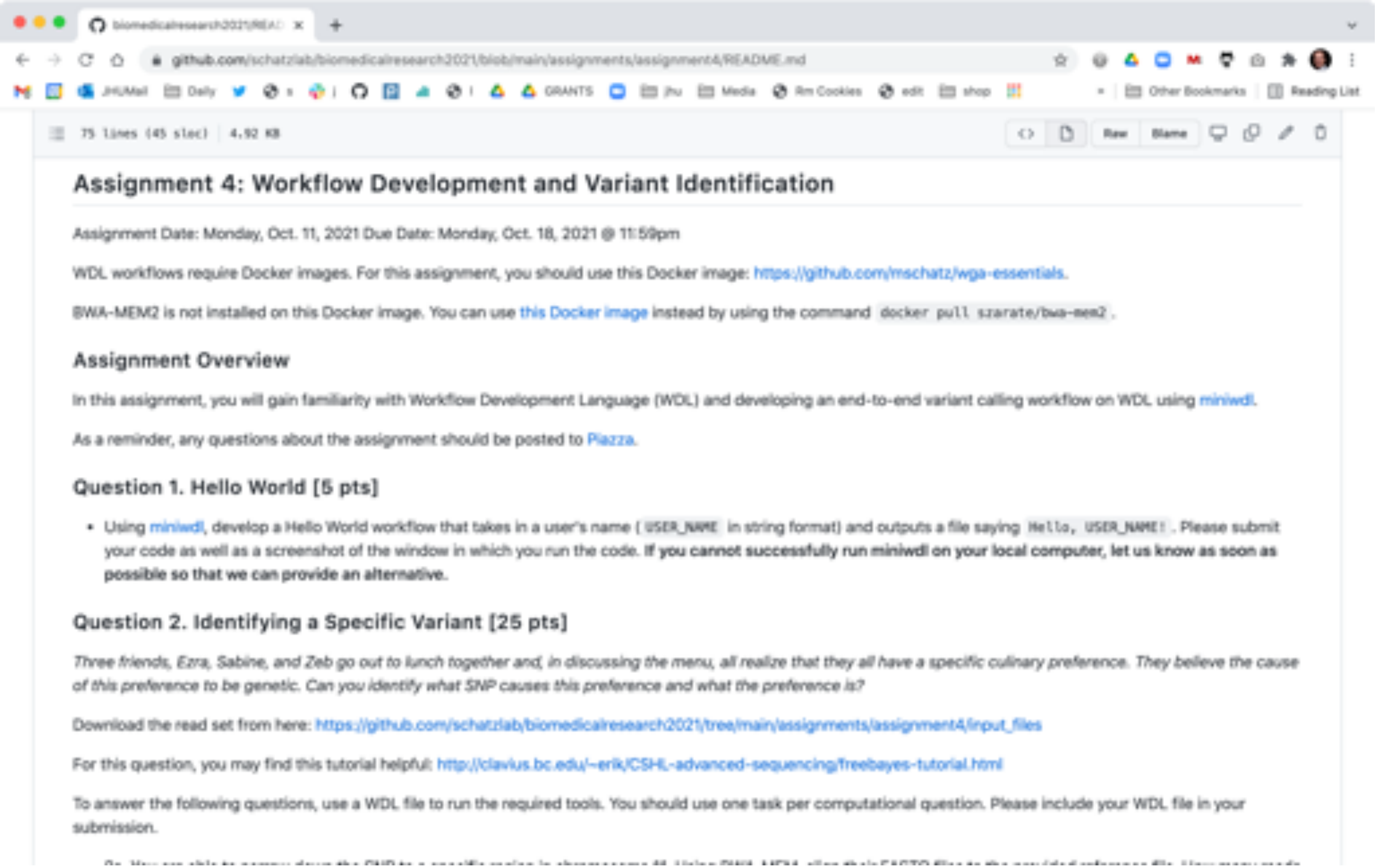
# Lecture 13. Gene Finding & RNAseq

Michael Schatz

October 13, 2021

Advanced Biomedical Research

# Assignment 4: WDLs
# Due Oct 18 @ 11:59pm


biomedicalresearch2021/READ
github.com/schatzlab/biomedicalresearch2021/blob/main/assignments/assignment4/README.md
JHUMail
Daily
GRANTS
jhu
Media
Rm Cookies
edit
shop
Other Bookmarks
Reading List
75 lines (45 sloc) | 4.92 KB
Raw
Blame

## Assignment 4: Workflow Development and Variant Identification

Assignment Date: Monday, Oct. 11, 2021 Due Date: Monday, Oct. 18, 2021 @ 11:59pm

WDL workflows require Docker images. For this assignment, you should use this Docker image: https://github.com/mschatz/wga-essentials.

BWA-MEM2 is not installed on this Docker image. You can use this Docker image instead by using the command `docker pull szarate/bwa-mem2`.

### Assignment Overview

In this assignment, you will gain familiarity with Workflow Development Language (WDL) and developing an end-to-end variant calling workflow on WDL using miniwdl.

As a reminder, any questions about the assignment should be posted to Piazza.

### Question 1. Hello World [5 pts]

- Using miniwdl, develop a Hello World workflow that takes in a user's name (`USER_NAME` in string format) and outputs a file saying `Hello, USER_NAME!`. Please submit your code as well as a screenshot of the window in which you run the code. **If you cannot successfully run miniwdl on your local computer, let us know as soon as possible so that we can provide an alternative.**

### Question 2. Identifying a Specific Variant [25 pts]

*Three friends, Ezra, Sabine, and Zeb go out to lunch together and, in discussing the menu, all realize that they all have a specific culinary preference. They believe the cause of this preference to be genetic. Can you identify what SNP causes this preference and what the preference is?*

Download the read set from here: https://github.com/schatzlab/biomedicalresearch2021/tree/main/assignments/assignment4/input_files

For this question, you may find this tutorial helpful: http://clavius.bc.edu/~erik/CSHL-advanced-sequencing/freebayes-tutorial.html

To answer the following questions, use a WDL file to run the required tools. You should use one task per computational question. Please include your WDL file in your submission.

https://github.com/schatzlab/biomedicalresearch2021

# Goal: Genome Annotations

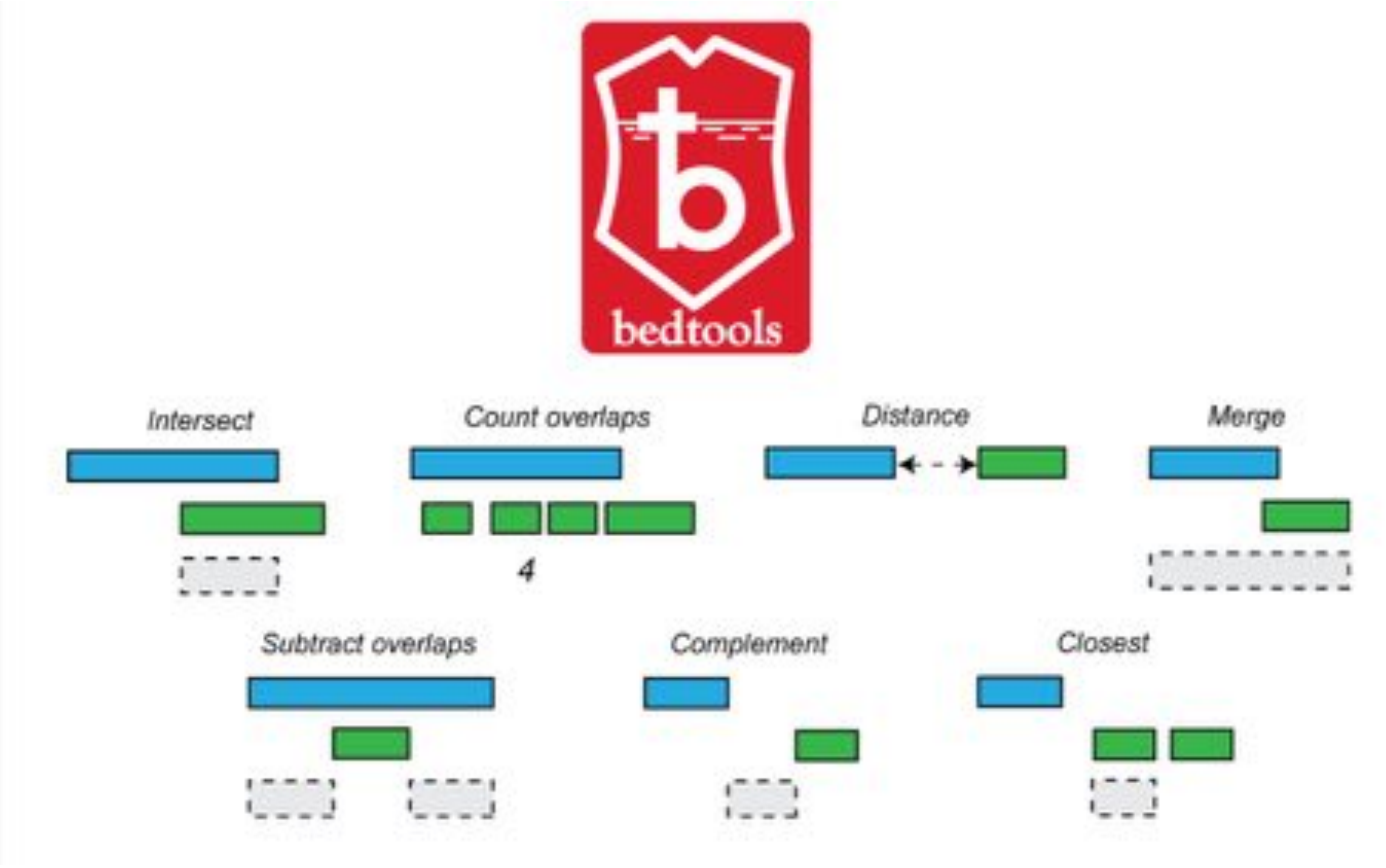
aatgcatgcggctatgctaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaat
gcatgcggctatgcaagctgggatccgatgactatgctaagctgggatccgatgacaatgcatgcggctatgct
aatgaatggtcttgggatttaccttggaatgctaagctgggatccgatgacaatgcatgcggctatgctaatgaa
tggtcttgggatttaccttggaatatgctaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcg
gctatgctaatgcatgcggctatgcaagctgggatccgatgactatgctaagctgcggctatgctaatgcatgcg
gctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctatgcaagctgggatcct
gcggctatgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatgacaatgcatgcggct
atgctaatgaatggtcttgggatttaccttggaatatgctaatgcatgcggctatgctaagctgggaatgcatgcg
gctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctatgcaagctgggatccg
atgactatgctaagctgcggctatgctaatgcatgcggctatgctaagctcatgcggctatgctaagctgggaat
gcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctatgcaagctg
ggatccgatgactatgctaagctgcggctatgctaatgcatgcggctatgctaagctcggctatgctaatgaatg
gtcttgggatttaccttggaatgctaagctgggatccgatgacaatgcatgcggctatgctaatgaatggtcttgg
gatttaccttggaatatgctaatgcatgcggctatgctaagctgggaatgcatgcggctatgctaagctgggatc
cgatgacaatgcatgcggctatgctaatgcatgcggctatgcaagctgggatccgatgactatgctaagctgcg
gctatgctaatgcatgcggctatgctaagctcatgcgg

# Goal: Genome Annotations

aatgcatgcggctatgctaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaat
gcatgcggctatgcaagctgggatccgatgactatgctaagctgggatccgatgacaatgcatgcggctatgct
aatgaatggtcttgggatttaccttggaatgctaagctgggatccgatgacaatgcatgcggctatgctaatgaa
tggtcttgggatttaccttggaatatgctaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcg
gctatgctaatgcatgcggctatgcaagctgggatccgatgactatgctaagctgcggctatgctaatgcatgcg
gctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctatgcaagctgggatcct
gcggctatgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatgacaatgcatgcggct
atgctaatgaatggtcttgggatt gctatgctaagctgggaatgcatgcg
gctatgctaagctgggatccgat atgcggctatgcaagctgggatccg
atgactatgctaagctgcggctatgctaatgcatgcggctatgctaagctcatgcggctatgctaagctgggaat
gcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctatgcaagctg
ggatccgatgactatgctaagctgcggctatgctaatgcatgcggctatgctaagctcggctatgctaatgaatg
gtcttgggatttaccttggaatgctaagctgggatccgatgacaatgcatgcggctatgctaatgaatggtcttgg
gatttaccttggaatatgctaatgcatgcggctatgctaagctgggaatgcatgcggctatgctaagctgggatc
cgatgacaatgcatgcggctatgctaatgcatgcggctatgcaagctgggatccgatgactatgctaagctgcg
gctatgctaatgcatgcggctatgctaagctcatgcgg

**Gene!**

# BEDTools to the rescue!



# Outline

1. Alignment to other genomes
2. Prediction aka “Gene Finding”
3. Experimental & Functional Assays

# Outline

1. Alignment to other genomes
2. Prediction aka “Gene Finding”
3. Experimental & Functional Assays

# Basic Local Alignment Search Tool

- Rapidly compare a sequence Q to a database to find all sequences in the database with an score above some cutoff S.
  - Which protein is most similar to a newly sequenced one?
  - Where does this sequence of DNA originate?

- Speed achieved by using a procedure that typically finds "most" matches with scores > S.
  - Tradeoff between sensitivity and specificity/speed
    - Sensitivity – ability to find all related sequences
    - Specificity – ability to reject unrelated sequences

(Altschul et al. 1990)

# Seed and Extend

```
FAKDFLAGGVAAAISKTAVAPIERVKLLLQVQHASKQITADKQYKGIIDCVVRIPKEQGV

FLIDLASGGTAAAVSKTAVAPIERVKLLLQVQDASKAIAVDKRYKGIMDVLIRVPKEQGV
```

- Homologous sequences are likely to contain a short high scoring word pair, a seed.
  - Smaller seed sizes make the sense more sensitive, but also (much) slower
  - Typically do a fast search for prototypes, but then most sensitive for final result

- BLAST then tries to extend high scoring word pairs to compute high scoring segment pairs (HSPs).
  - Significance of the alignment reported via an e-value

# Seed and Extend

```
FAKDFLAGGVAAAISKTAVAPIERVKLLLQVQHASKQITADKQYKGIIDCVVRIPKEQGV
|  |    | ||| |||||||||||||||||| ||| |  |  |||| |   | ||||||
FLIDLASGGTAAAVSKTAVAPIERVKLLLQVQDASKAIAVDKRYKGIMDVLIRVPKEQGV
```

- Homologous sequences are likely to contain a short high scoring word pair, a seed.
  - Smaller seed sizes make the sense more sensitive, but also (much) slower
  - Typically do a fast search for prototypes, but then most sensitive for final result

- BLAST then tries to extend high scoring word pairs to compute high scoring segment pairs (HSPs).
  - Significance of the alignment reported via an e-value

# BLAST E-values

E-value = the number of HSPs having alignment score S (or higher) expected to occur by chance.

→ Smaller E-value, more significant in statistics

→ Bigger E-value, less significant

→ Over 1 means expect this totally by chance (not significant at all!)

The expected number of HSPs with the score at least S is :

$$E = K*n*m*e^{-\lambda S}$$

$K, \lambda$ are constant depending on model

$n, m$ are the length of query and sequence

E-values quickly drop off for better alignment bits scores

# Very Similar Sequences

```
Query: HBA_HUMAN Hemoglobin alpha subunit
Sbjct: HBB_HUMAN Hemoglobin beta subunit

Score =  114 bits (285),  Expect = 1e-26
Identities = 61/145 (42%), Positives = 86/145 (59%), Gaps = 8/145 (5%)

Query  2   LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF------DLSHGSAQV 55
           L+P +K+ V A WGKV  +  E G EAL R+ + +P T+ +F  F      D   G+ +V
Sbjct  3   LTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKV 60

Query  56  KGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPA 115
           K HGKKV  A ++ +AH+D++    + LS+LH  KL VDP NF+LL + L+  LA H
Sbjct  61  KAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGK 120

Query  116 EFTPAVHASLDKFLASVSTVLTSKY 140
           EFTP V A+  K +A V+  L  KY
Sbjct  121 EFTPPVQAAYQKVVAGVANALAHKY 145
```

# Quite Similar Sequences

```
Query: HBA_HUMAN Hemoglobin alpha subunit
Sbjct: MYG_HUMAN Myoglobin

Score = 51.2 bits (121), Expect = 1e-07,
Identities = 38/146 (26%), Positives = 58/146 (39%), Gaps = 6/146 (4%)

Query  2  LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF------DLSHGSAQV  55
          LS  +   V   WGKV A    +G E L R+F   P T   F  F      D    S  +
Sbjct  3  LSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASEDL  62

Query  56 KGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPA  115
          K HG  V  AL   +         +  L+  HA K ++     + +S C++  L +  P
Sbjct  63 KKHGATVLTALGGILKKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKHPG  122

Query  116 EFTPAVHASLDKFLASVSTVLTSKYR  141
           +F      +++K L      + S Y+
Sbjct  123 DFGADAQGAMNKALELFRKDMASNYK  148
```

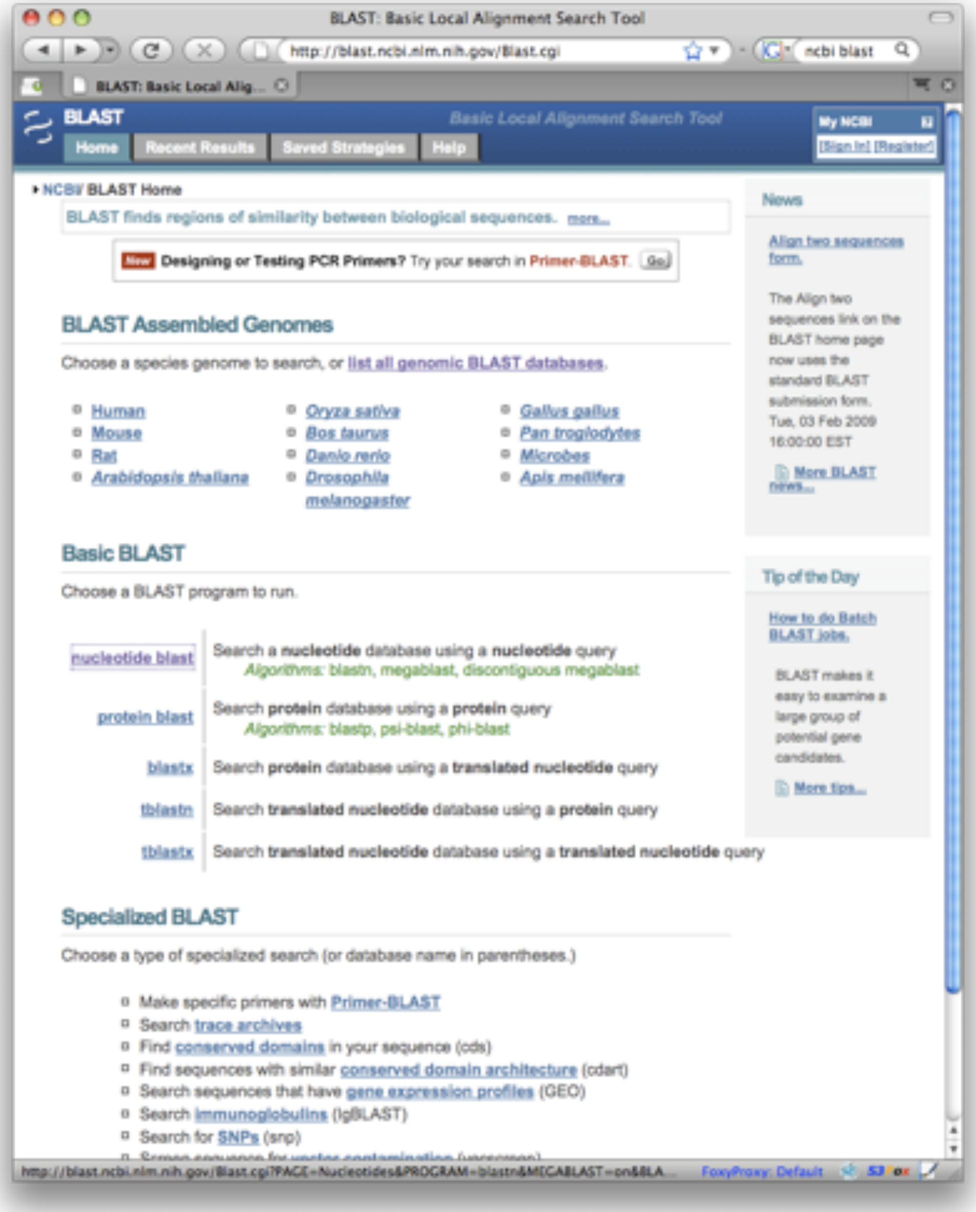
# Not similar sequences

```
Query: HBA_HUMAN Hemoglobin alpha subunit
Sbjct: SPAC869.02c [Schizosaccharomyces pombe]

 Score = 33.1 bits (74),  Expect = 0.24
 Identities = 27/95 (28%), Positives = 50/95 (52%), Gaps = 10/95 (10%)

Query  30  ERMFLSFPTTKTYFPHFDLSHGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAH  89
           ++M  ++P      P+F+ +H  +      + +A AL N   ++DD+  +LSA  D
Sbjct  59  QKMLGNYPEV---LPYFNKAHQISL--SQPRILAFALLNYAKNIDDL-TSLSAFMDQIVV 112

Query  90  K---LRVDPVNFKLLSHCLLVTLAAHLPAEF-TPA  120
           K   L++   ++ ++ HCLL T+   LP++  TPA
Sbjct  113 KHVGLQIKAEHYPIVGHCLLSTMQELLPSDVATPA  147
```

# Blast Versions

| Program | Database | Query |
|---|---|---|
| BLASTN | Nucleotide | Nucleotide |
| BLASTP | Protein | Protein |
| BLASTX | Protein | Nucleotide translated into protein |
| TBLASTN | Nucleotide translated into protein | Protein |
| TBLASTX | Nucleotide translated into protein | Nucleotide translated into protein |

# NCBI Blast


BLAST: Basic Local Alignment Search Tool
http://blast.ncbi.nlm.nih.gov/Blast.cgi
ncbi blast
BLAST: Basic Local Alig...
BLAST
Basic Local Alignment Search Tool
Home
Recent Results
Saved Strategies
Help
My NCBI
[Sign In] [Register]

NCBI/ BLAST Home

BLAST finds regions of similarity between biological sequences. more...

New Designing or Testing PCR Primers? Try your search in Primer-BLAST. Go

BLAST Assembled Genomes

Choose a species genome to search, or list all genomic BLAST databases.

- Human
- Mouse
- Rat
- Arabidopsis thaliana
- *Oryza sativa*
- *Bos taurus*
- *Danio rerio*
- *Drosophila melanogaster*
- *Gallus gallus*
- *Pan troglodytes*
- *Microbes*
- *Apis mellifera*

Basic BLAST

Choose a BLAST program to run.

nucleotide blast: Search a **nucleotide** database using a **nucleotide** query
*Algorithms:* blastn, megablast, discontiguous megablast

protein blast: Search **protein** database using a **protein** query
*Algorithms:* blastp, psi-blast, phi-blast

blastx: Search **protein** database using a **translated nucleotide** query

tblastn: Search **translated nucleotide** database using a **protein** query

tblastx: Search **translated nucleotide** database using a **translated nucleotide** query

Specialized BLAST

Choose a type of specialized search (or database name in parentheses.)

- Make specific primers with Primer-BLAST
- Search trace archives
- Find conserved domains in your sequence (cds)
- Find sequences with similar conserved domain architecture (cdart)
- Search sequences that have gene expression profiles (GEO)
- Search immunoglobulins (IgBLAST)
- Search for SNPs (snp)
- Screen sequence for vector contamination (vecscreen)

News

Align two sequences form.

The Align two sequences link on the BLAST home page now uses the standard BLAST submission form.
Tue, 03 Feb 2009 16:00:00 EST

More BLAST news...

Tip of the Day

How to do Batch BLAST jobs.

BLAST makes it easy to examine a large group of potential gene candidates.

More tips...

http://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE=Nucleotides&PROGRAM=blastn&MEGABLAST=on&BLA...

- Nucleotide Databases
  - nr: All Genbank
  - refseq: Reference organisms
  - wgs: All reads
- Protein Databases
  - nr: All non-redundant sequences
  - Refseq: Reference proteins

# Outline

1. Alignment to other genomes
2. Prediction aka “Gene Finding”
3. Experimental & Functional Assays

# Bacterial Gene Finding and Glimmer

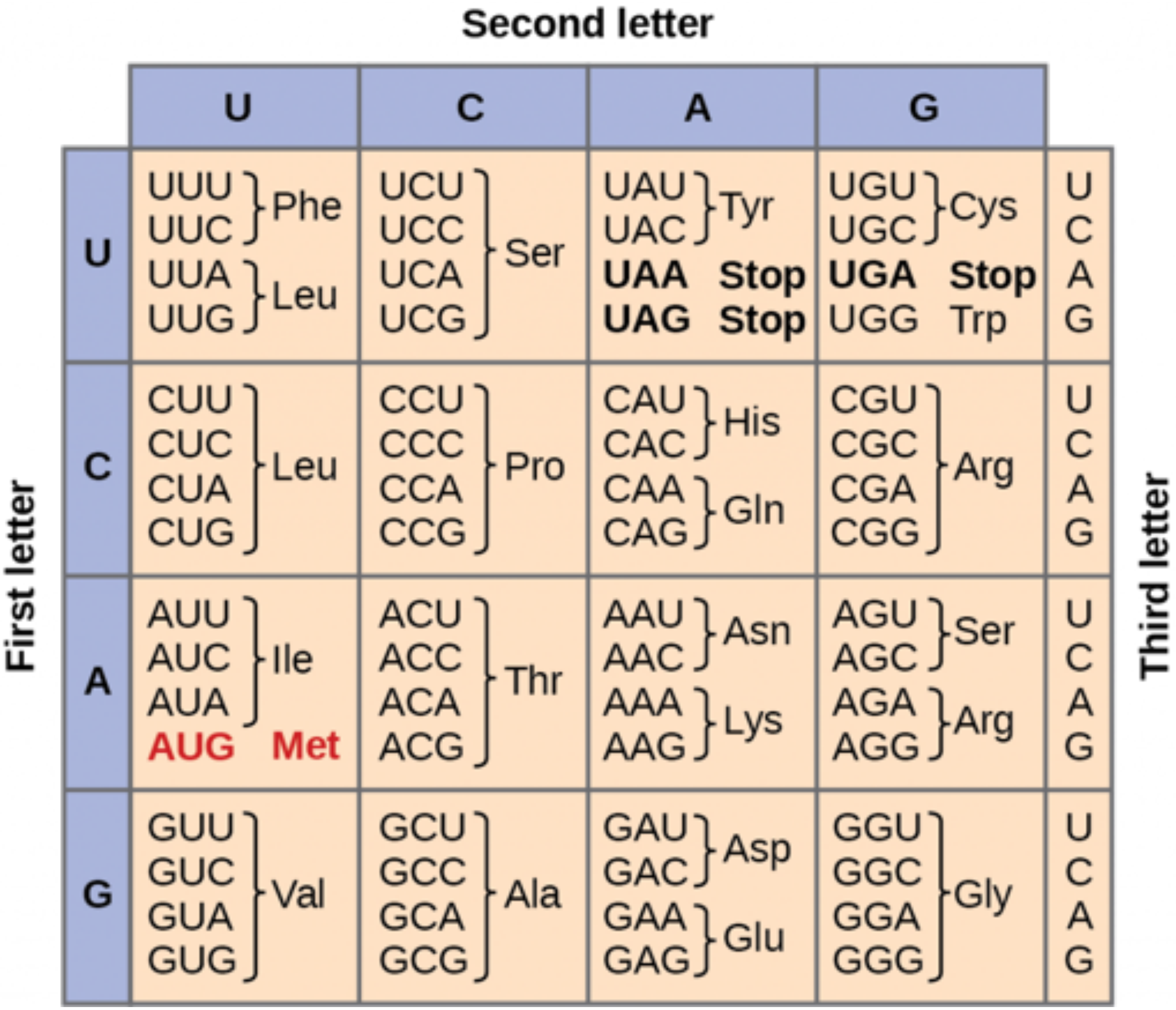
(also Archaeal and viral gene finding)

Arthur L. Delcher and Steven Salzberg

Center for Bioinformatics and Computational Biology

Johns Hopkins University

# Genetic Code



|  | Second letter |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| First letter | U |  | C |  | A |  | G |  | Third letter |
| U | UUU | Phe | UCU | Ser | UAU | Tyr | UGU | Cys | U |
|  | UUC |  | UCC |  | UAC |  | UGC |  | C |
|  | UUA | Leu | UCA |  | **UAA** | **Stop** | **UGA** | **Stop** | A |
|  | UUG |  | UCG |  | **UAG** | **Stop** | UGG | Trp | G |
| C | CUU | Leu | CCU | Pro | CAU | His | CGU | Arg | U |
|  | CUC |  | CCC |  | CAC |  | CGC |  | C |
|  | CUA |  | CCA |  | CAA | Gln | CGA |  | A |
|  | CUG |  | CCG |  | CAG |  | CGG |  | G |
| A | AUU | Ile | ACU | Thr | AAU | Asn | AGU | Ser | U |
|  | AUC |  | ACC |  | AAC |  | AGC |  | C |
|  | AUA |  | ACA |  | AAA | Lys | AGA | Arg | A |
|  | AUG | Met | ACG |  | AAG |  | AGG |  | G |
| G | GUU | Val | GCU | Ala | GAU | Asp | GGU | Gly | U |
|  | GUC |  | GCC |  | GAC |  | GGC |  | C |
|  | GUA |  | GCA |  | GAA | Glu | GGA |  | A |
|  | GUG |  | GCG |  | GAG |  | GGG |  | G |

Start:

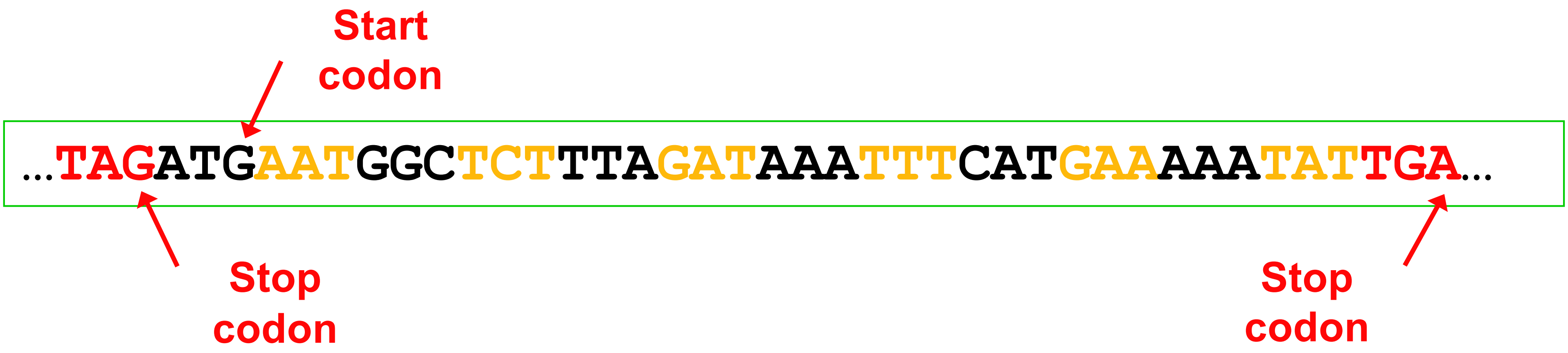
- AUG

Stop:

- UAA
- UAG
- UGA

# Step One

- Find open reading frames (ORFs).



Start codon

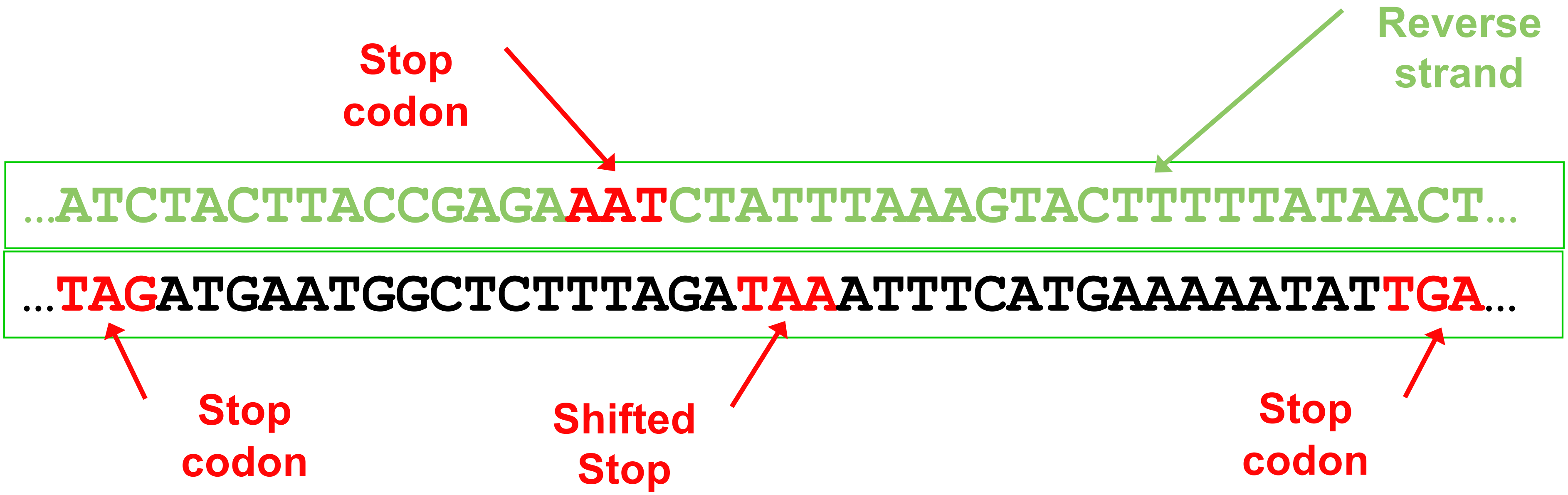
...TAGATGAATGGCTCTTTAGATAAATTTCATGAAAAATATTGA...

Stop codon

Stop codon

# Step One

- Find open reading frames (ORFs).



Stop codon

Reverse strand

...ATCTACTTACCGAGAAATCTATTTAAAGTACTTTTTATAACT...

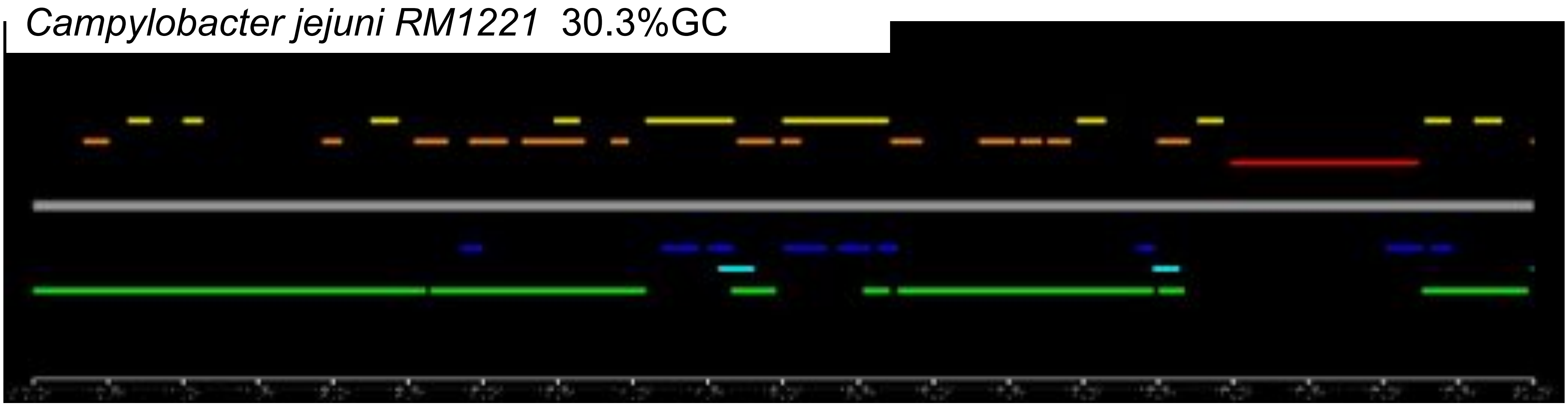
...TAGATGAATGGCTCTTTAGATAAATTTCATGAAAAATATTGA...

Stop codon

Shifted Stop

Stop codon

- But ORFs generally overlap …



*Campylobacter jejuni RM1221* 30.3%GC

All ORFs longer than 100bp on both strands shown

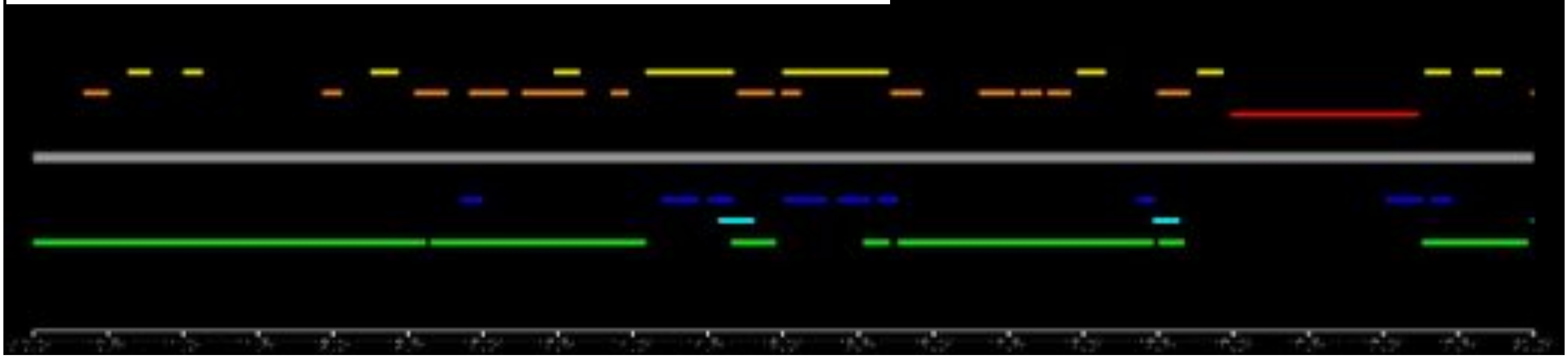
- color indicates reading frame

Longest ORFs likely to be protein-coding genes

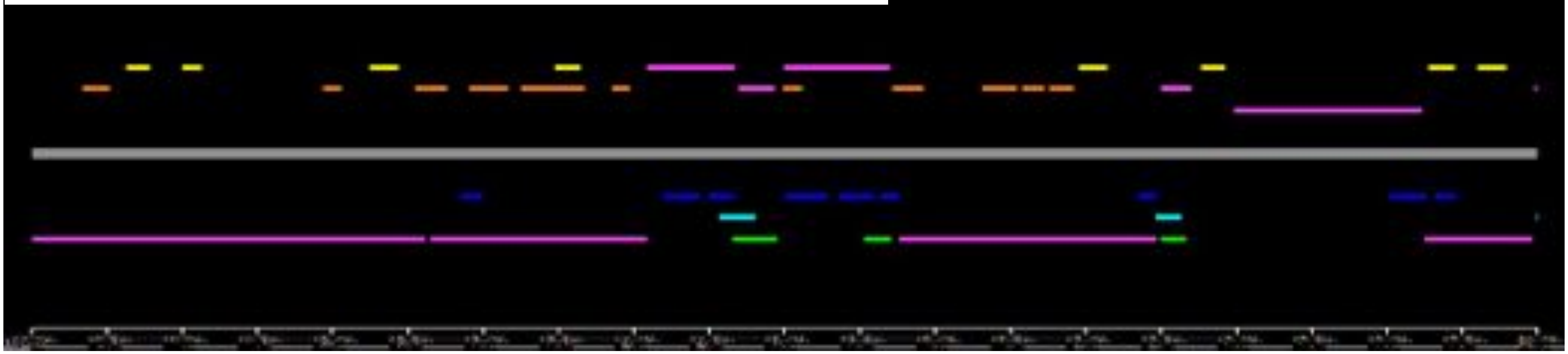
Note the low GC content

All genes are ORFs but not all ORFs are genes

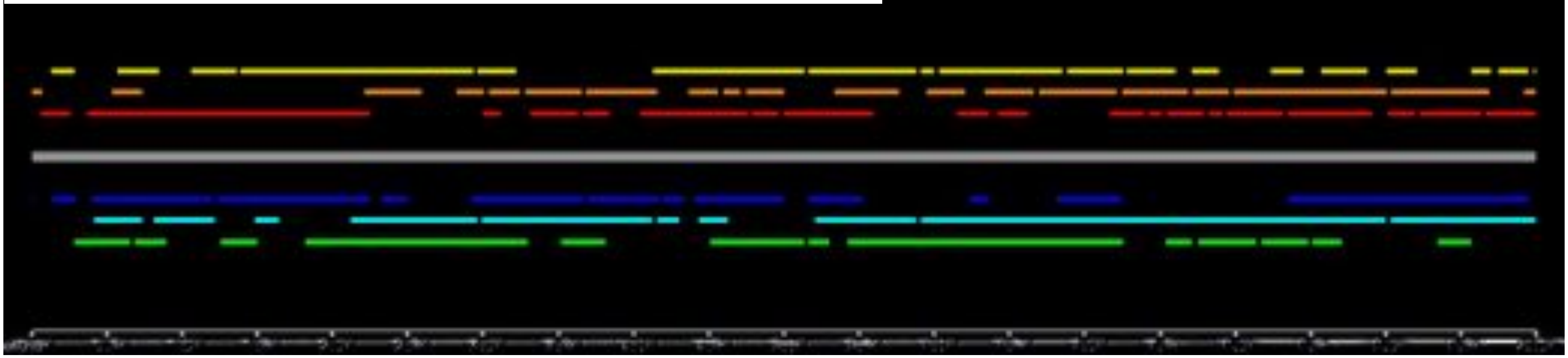
*Campylobacter jejuni RM1221* 30.3%GC



*Campylobacter jejuni RM1221* 30.3%GC

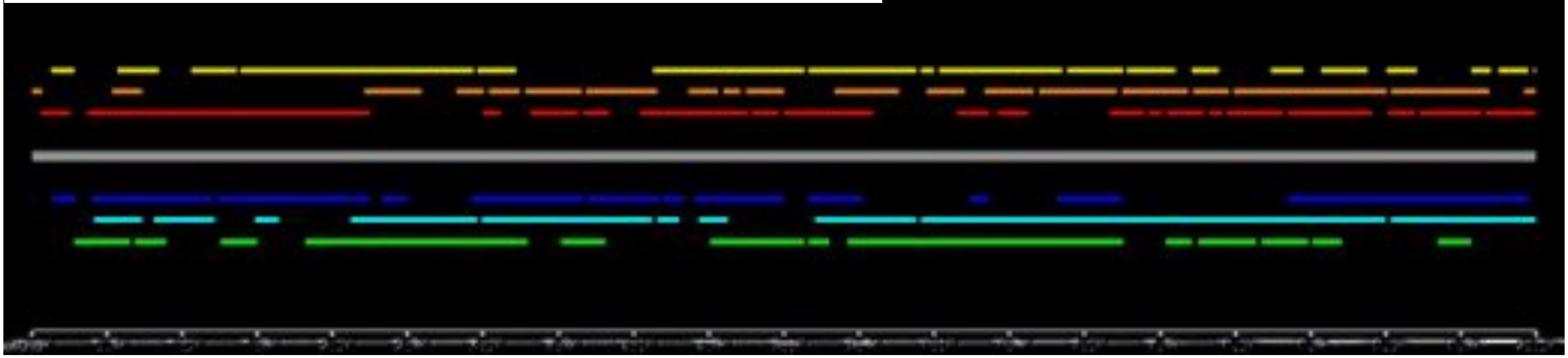


*Mycobacterium smegmatis MC2*  67.4%GC

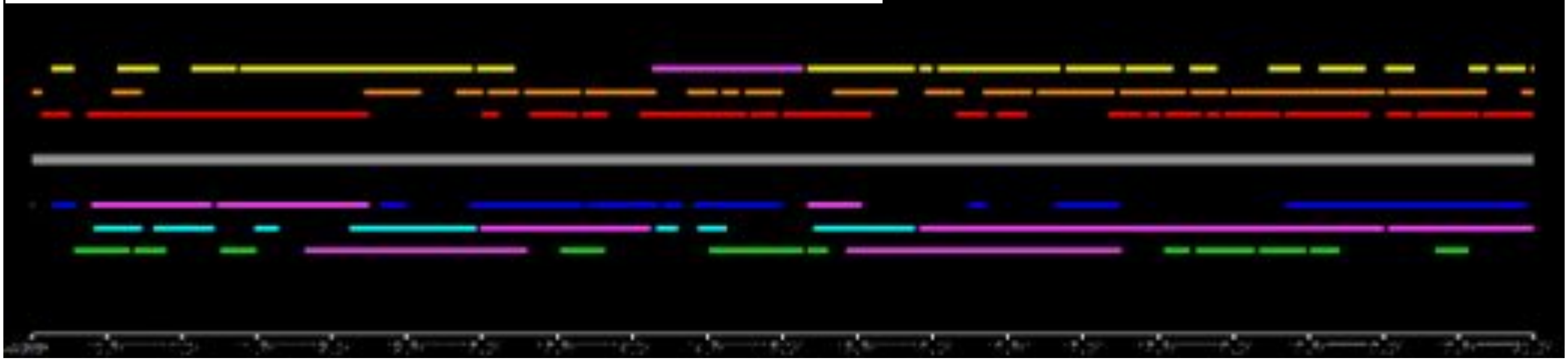


Note what happens in a high-GC genome

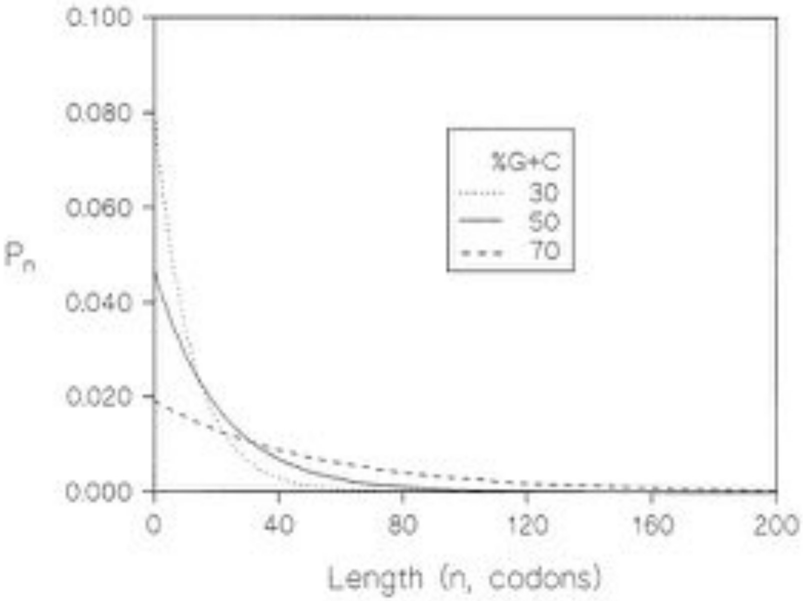
*Mycobacterium smegmatis MC2* 67.4%GC

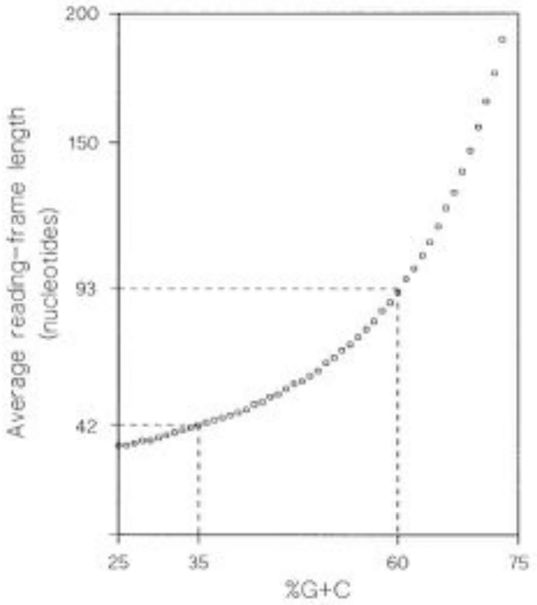


*Mycobacterium smegmatis MC2* 67.4%GC



# Stop Codon Frequencies





***If the sequence is mostly A+T, then likely to form stop codons by chance!***

***In High A+T (Low G+C):***
Frequent stop codons; Short Random ORFs; long ORFs likely to be true genes

***In High G+C (Low A+T):***
Rare stop codons; Long Random ORFs; harder to identify true genes

***A relationship between GC content and coding-sequence length.***
Oliver & Marín (1996) J Mol Evol. 43(3):216-23.

# Probabilistic Methods

- Create models that have a probability of generating any given sequence.
  - Evaluate gene/non-genome models against a sequence
- Train the models using examples of the types of sequences to generate.
  - Use RNA sequencing, homology, or "obvious" genes
- The "score" of an orf is the probability of the model generating it.
  - Most basic technique is to count how kmers occur in known genes versus intergenic sequences
  - More sophisticated methods consider variable length contexts, "wobble" bases, other statistical clues

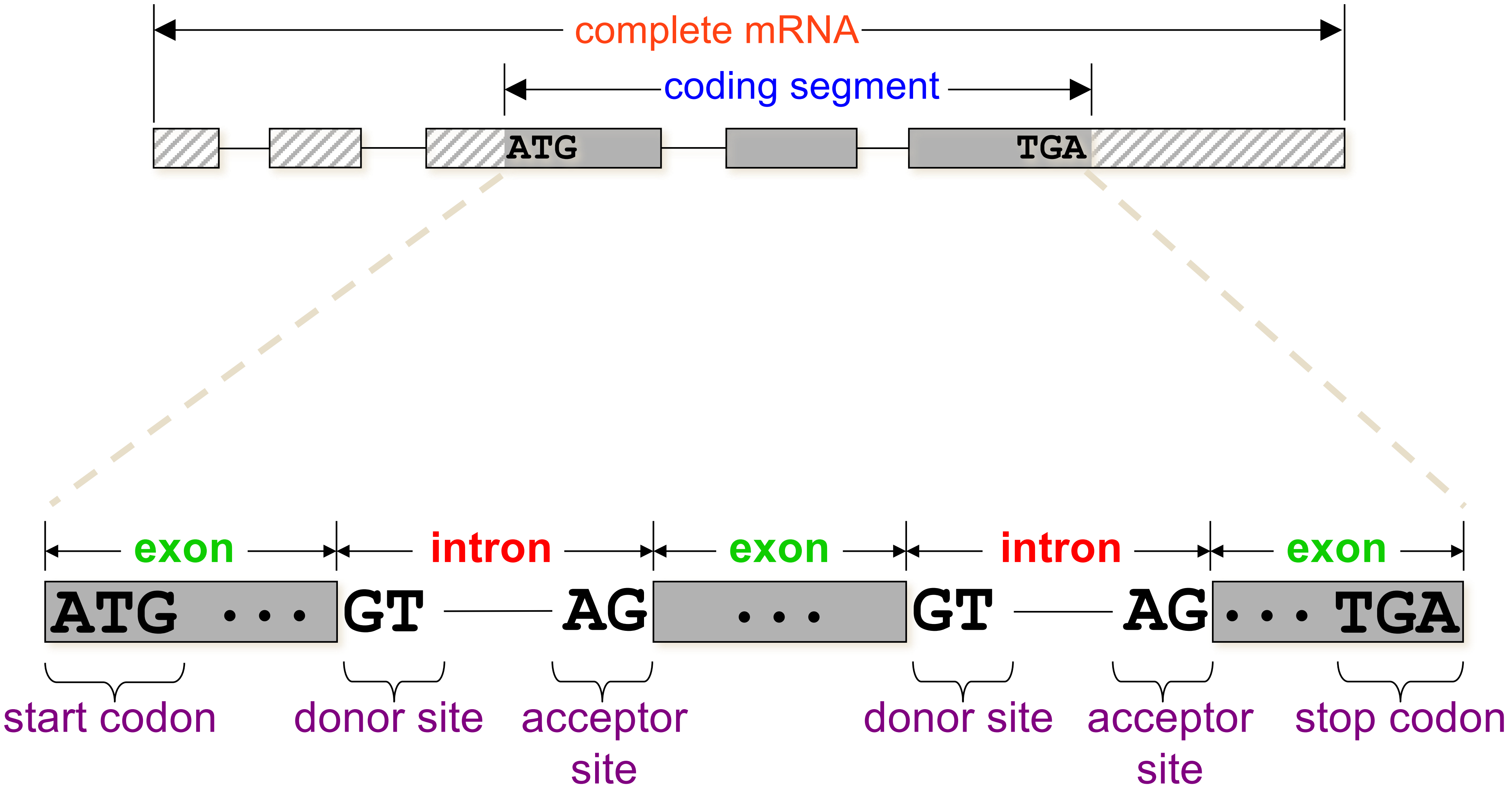
# Overview of Eukaryotic Gene Prediction

CBB 231 / COMPSCI 261

*W.H. Majoros*

Duke University

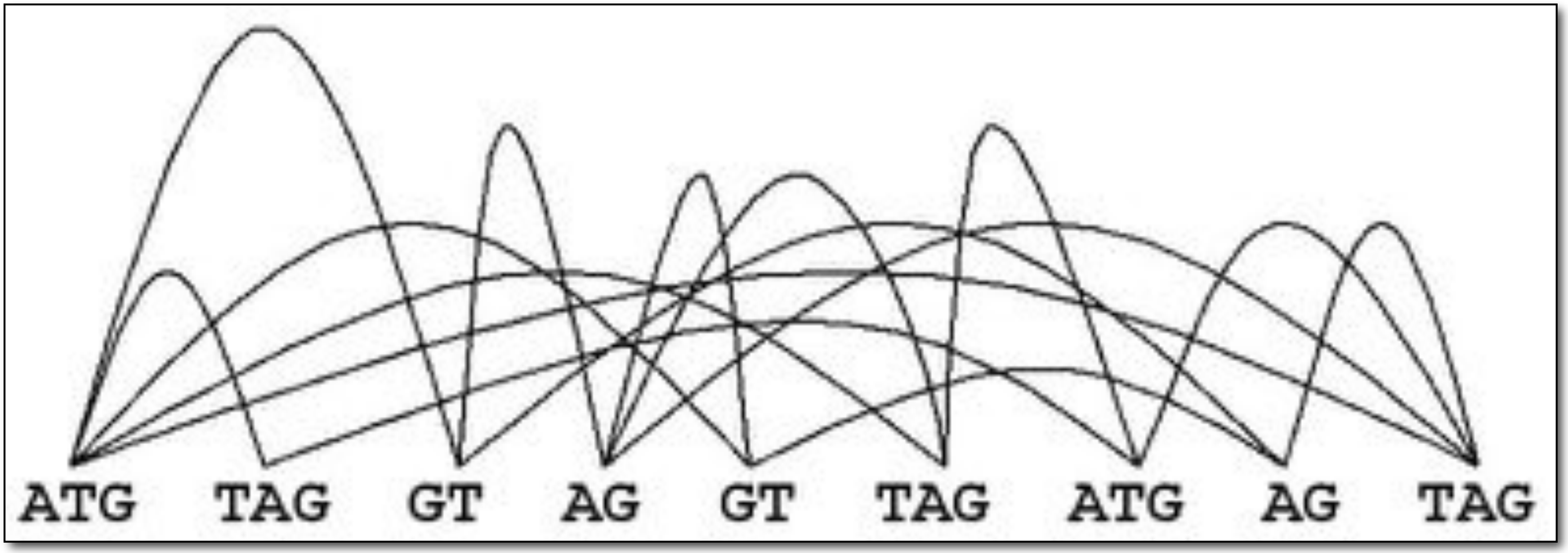
# Eukaryotic Gene Syntax



Regions of the gene outside of the CDS are called *UTR*'s (*untranslated regions*), and are mostly ignored by gene finders, though they are important for regulatory functions.

# Representing Gene Syntax with ORF Graphs

After identifying the most promising (i.e., highest-scoring) signals in an input sequence, we can apply the gene syntax rules to connect these into an *ORF graph*:

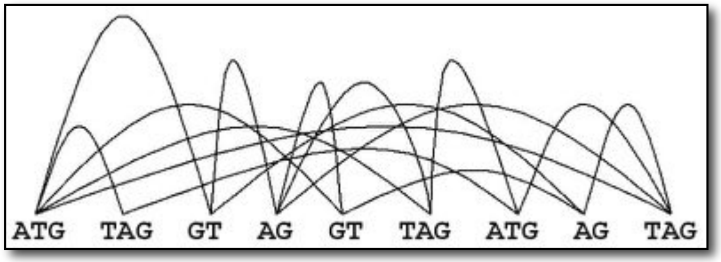


An ORF graph represents all possible *gene parses* (and their scores) for a given set of putative signals. A *path* through the graph represents a single gene parse.

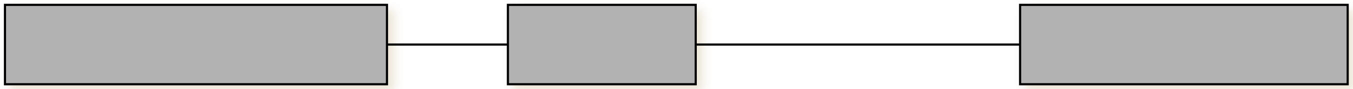
# Conceptual Gene-finding Framework

```
TATTCCGATCGATCGATCTCTCTAGCGTCTACG
CTATCATCGCTCTCTATTATCGCGCGATCGTCG
ATCGCGCGAGAGTATGCTACGTCGATCGAATTG
```

identify most promising signals, score signals and content regions between them; induce an ORF graph on the signals



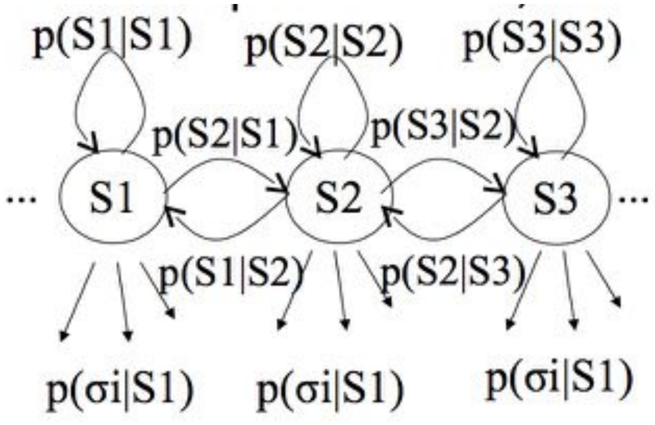
find highest-scoring path through ORF graph; interpret path as a gene parse = gene structure



Duke UNIVERSITY

# HMMs for Gene Finding

- Similar to Markov models used for prokaryotic gene finding, but system may transition between multiple models called states (gene/non-gene, intergenic/exon/intron)
- Observers can see the emitted symbols of an HMM (i.e., nucleotides) but have no ability to know which state the HMM is currently in.
  - But we can ***infer*** the most likely hidden states of an HMM based on the given sequence of emitted symbols.



AAAGCATGCATTTAACGTGAGCACAATAGATTACA

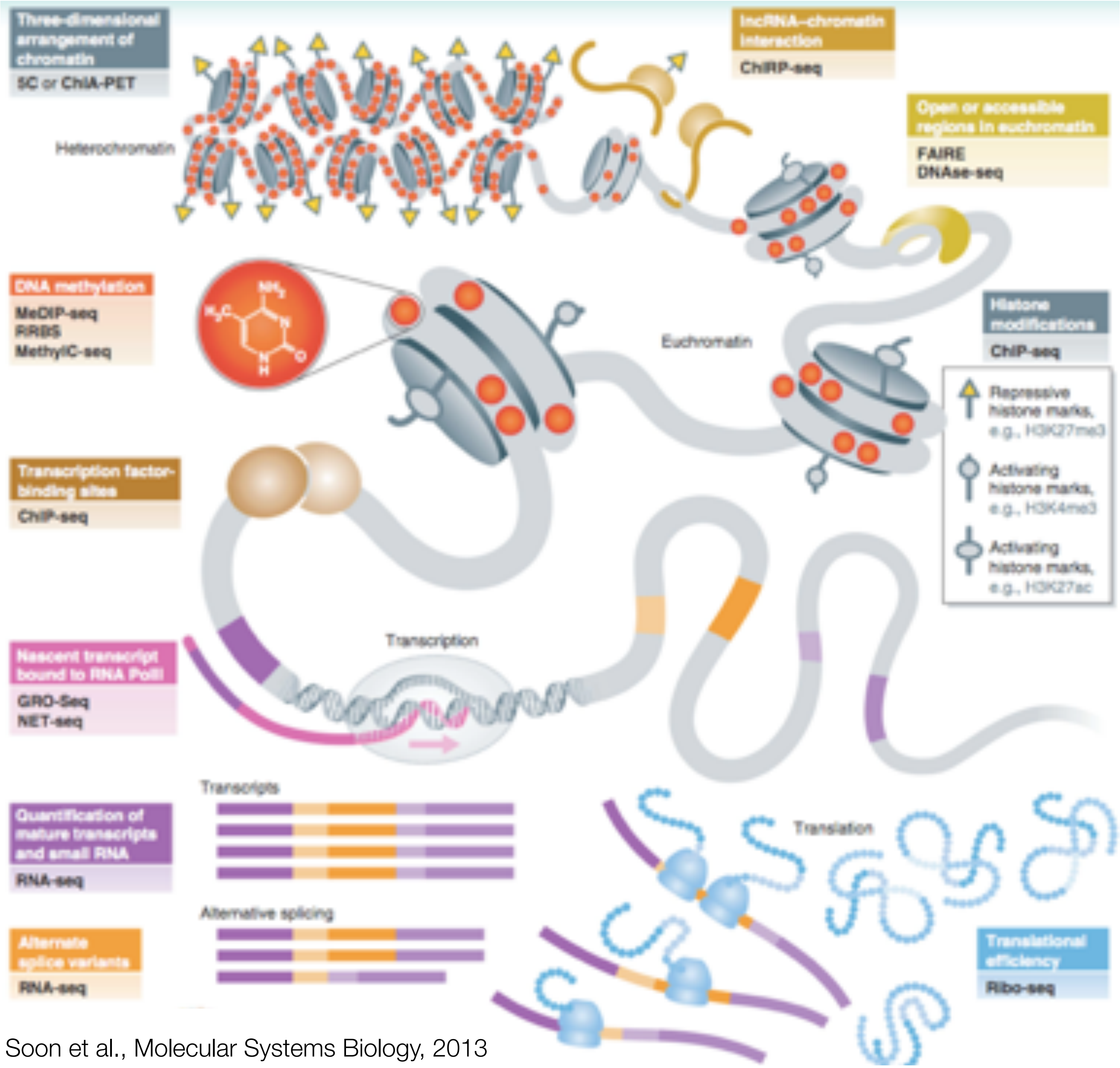
# Outline

1. Alignment to other genomes
2. Prediction aka “Gene Finding”
3. Experimental & Functional Assays

# Sequencing Assays

**The *Seq List (in chronological order)**

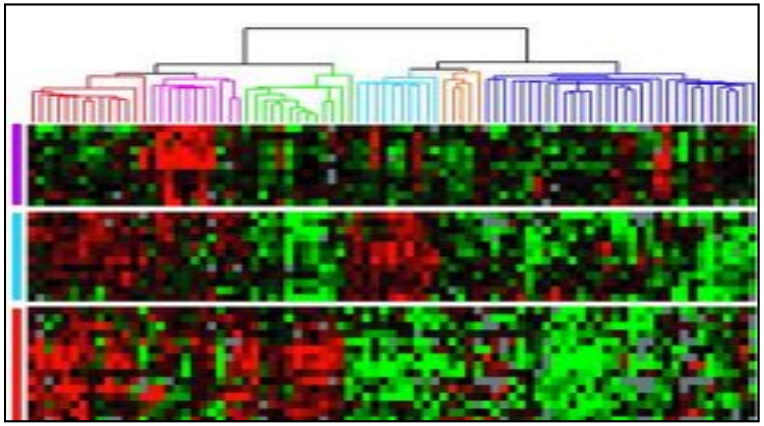
1. Gregory E. Crawford et al., "Genome-wide Mapping of DNase Hypersensitive Sites Using Massively Parallel Signature Sequencing (MPSS)," Genome Research 16, no. 1 (January 1, 2006): 123–131, doi:10.1101/gr.4074106.
2. David S. Johnson et al., "Genome-Wide Mapping of in Vivo Protein-DNA Interactions," Science 316, no. 5830 (June 8, 2007): 1497–1502, doi:10.1126/science.1141319.
3. Tarjei S. Mikkelsen et al., "Genome-wide Maps of Chromatin State in Pluripotent and Lineage-committed Cells," Nature 448, no. 7153 (August 2, 2007): 553–560, doi:10.1038/nature06008.
4. Thomas A. Down et al., "A Bayesian Deconvolution Strategy for Immunoprecipitation-based DNA Methylome Analysis," Nature Biotechnology 26, no. 7 (July 2008): 779–785, doi:10.1038/nbt1414.
5. Ali Mortazavi et al., "Mapping and Quantifying Mammalian Transcriptomes by RNA-Seq," Nature Methods 5, no. 7 (July 2008): 621–628, doi:10.1038/nmeth.1226.
6. Nathan A. Baird et al., "Rapid SNP Discovery and Genetic Mapping Using Sequenced RAD Markers," PLoS ONE 3, no. 10 (October 13, 2008): e3376, doi:10.1371/journal.pone.0003376.
7. Leighton J. Core, Joshua J. Waterfall, and John T. Lis, "Nascent RNA Sequencing Reveals Widespread Pausing and Divergent Initiation at Human Promoters," Science 322, no. 5909 (December 19, 2008): 1845–1848, doi:10.1126/science.1162228.
8. Chao Xie and Martti T. Tammi, "CNV-seq, a New Method to Detect Copy Number Variation Using High-throughput Sequencing," BMC Bioinformatics 10, no. 1 (March 6, 2009): 80, doi:10.1186/1471-2105-10-80.
9. Jay R. Hesselberth et al., "Global Mapping of protein-DNA Interactions in Vivo by Digital Genomic Footprinting," Nature Methods 6, no. 4 (April 2009): 283–289, doi:10.1038/nmeth.1313.
10. Nicholas T. Ingolia et al., "Genome-Wide Analysis in Vivo of Translation with Nucleotide Resolution Using Ribosome Profiling," Science 324, no. 5924 (April 10, 2009): 218–223, doi:10.1126/science.1168978.
11. Alayne L. Brunner et al., "Distinct DNA Methylation Patterns Characterize Differentiated Human Embryonic Stem Cells and Developing Human Fetal Liver," Genome Research 19, no. 6 (June 1, 2009): 1044–1056, doi:10.1101/gr.088773.108.
12. Mayumi Oda et al., "High-resolution Genome-wide Cytosine Methylation Profiling with Simultaneous Copy Number Analysis and Optimization for Limited Cell Numbers," Nucleic Acids Research 37, no. 12 (July 1, 2009): 3829–3839, doi:10.1093/nar/gkp260.
13. Zachary D. Smith et al., "High-throughput Bisulfite Sequencing in Mammalian Genomes," Methods 48, no. 3 (July 2009): 226–232, doi:10.1016/j.ymeth.2009.05.003.
14. Andrew M. Smith et al., "Quantitative Phenotyping via Deep Barcode Sequencing," Genome Research (July 21, 2009).


Three-dimensional arrangement of chromatin
5C or ChIA-PET
lncRNA–chromatin interaction
ChIRP-seq
Open or accessible regions in euchromatin
FAIRE
DNAse-seq
Heterochromatin
DNA methylation
MeDIP-seq
RRBS
MethylC-seq
$NH_2$
$H_3C$
N
N
H
O
Euchromatin
Histone modifications
ChIP-seq
Repressive histone marks, e.g., H3K27me3
Activating histone marks, e.g., H3K4me3
Activating histone marks, e.g., H3K27ac
Transcription factor-binding sites
ChIP-seq
Transcription
Nascent transcript bound to RNA PolII
GRO-Seq
NET-seq
Transcripts
Quantification of mature transcripts and small RNA
RNA-seq
Alternative splicing
Alternate splice variants
RNA-seq
Translation
Translational efficiency
Ribo-seq

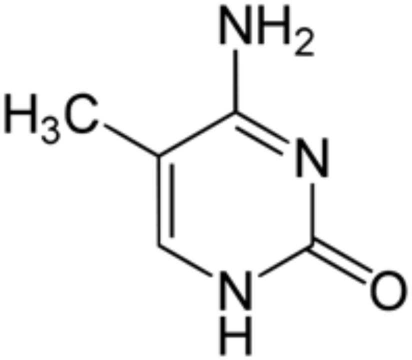
Soon et al., Molecular Systems Biology, 2013

# *-seq in 4 short vignettes

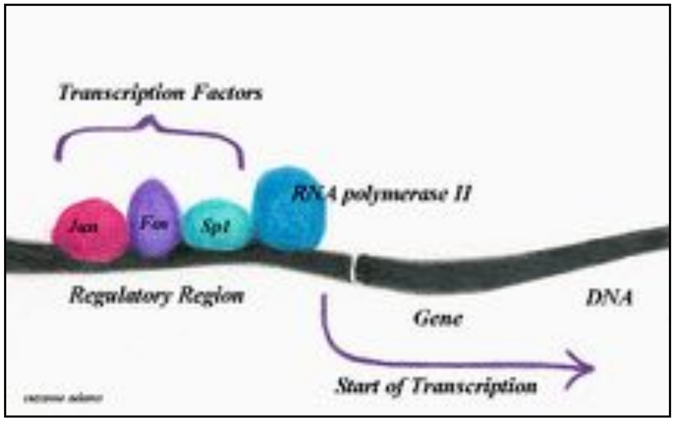
## RNA-seq



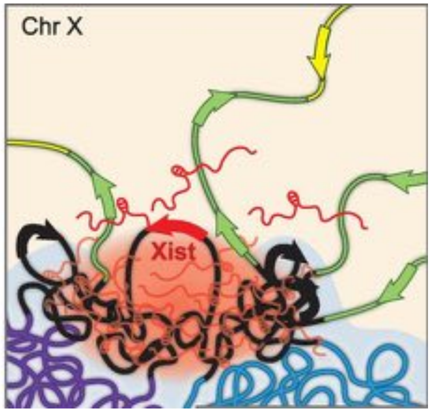
## Methyl-seq



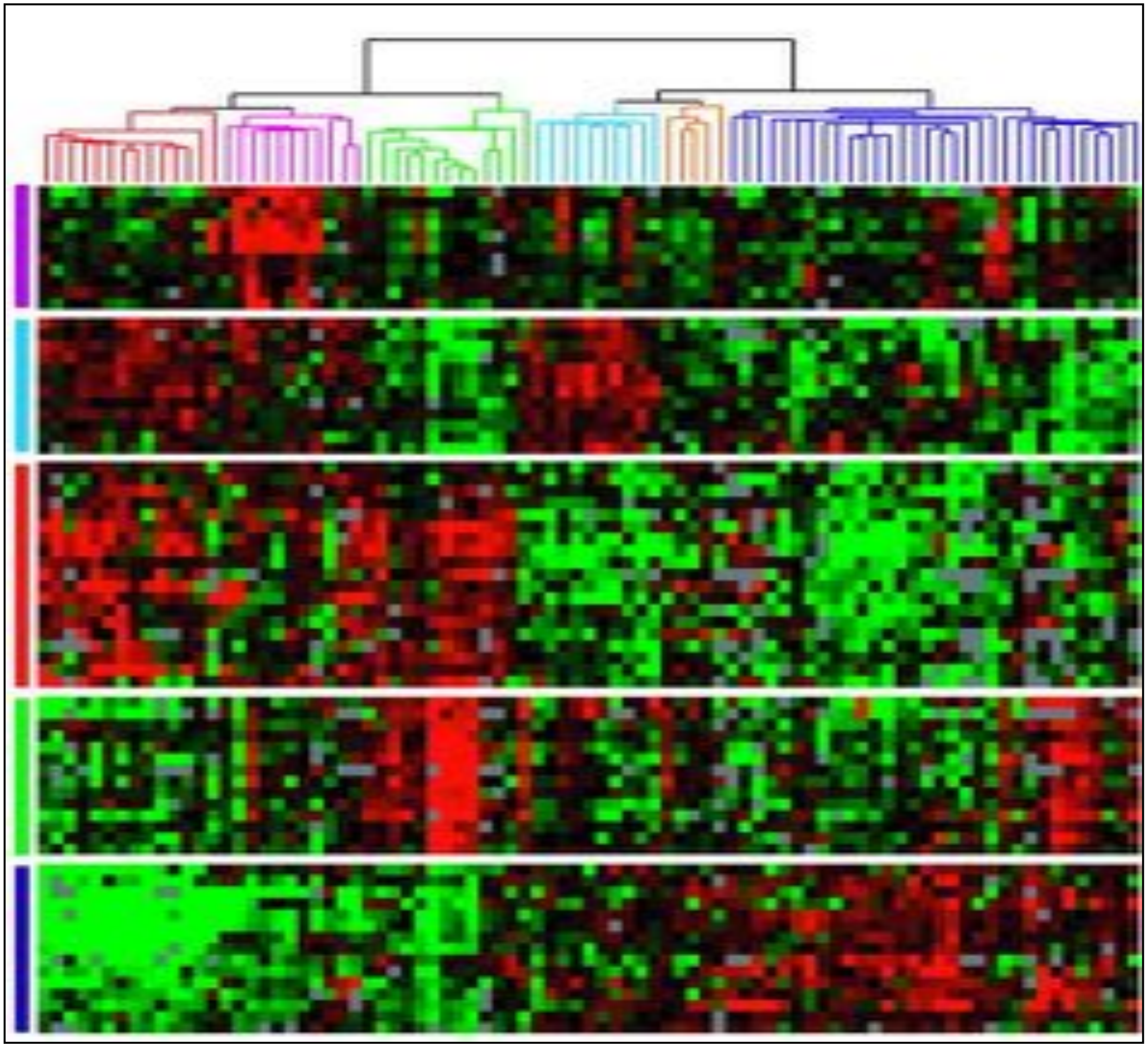
## ChIP-seq



## Hi-C

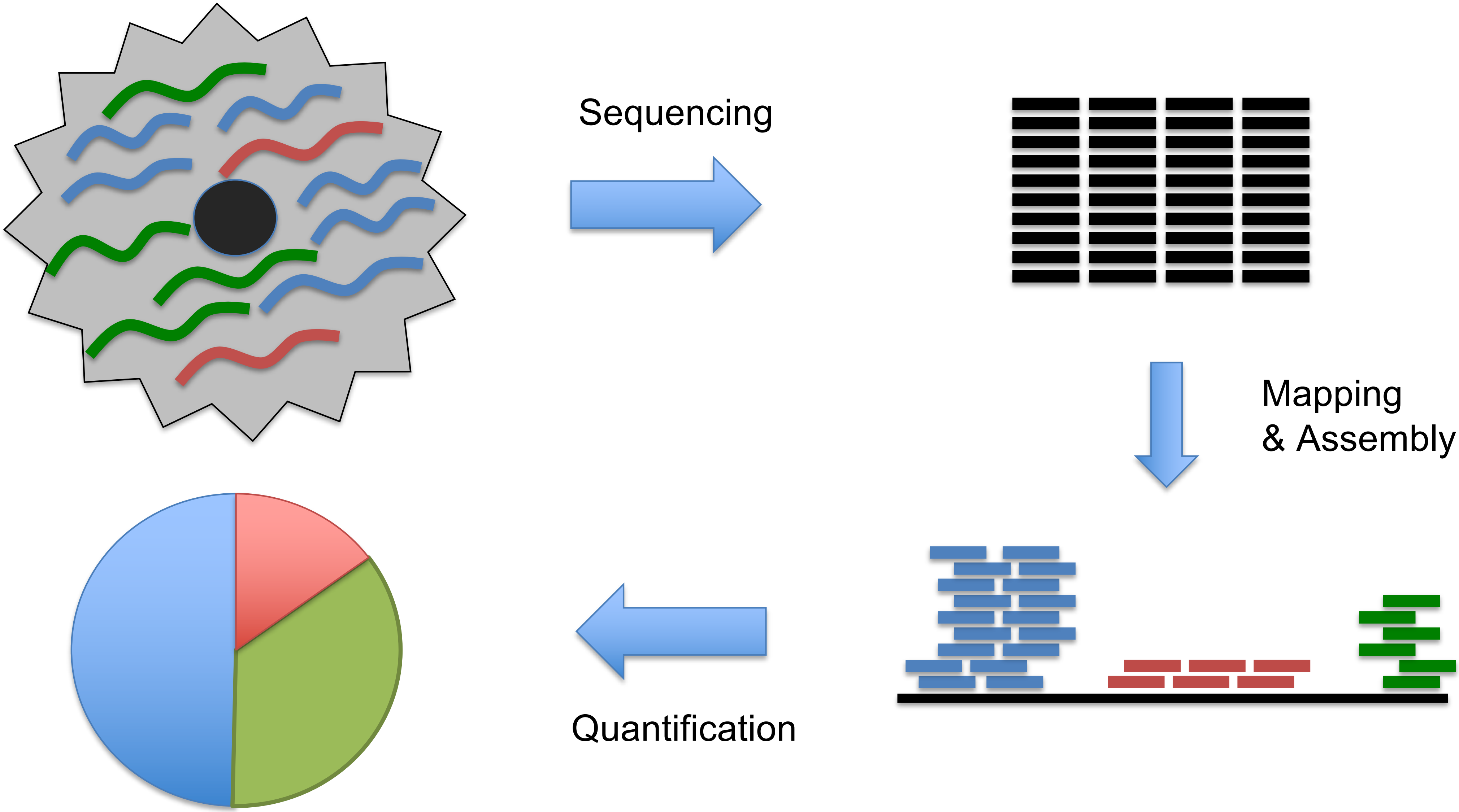


# RNA-seq



**Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications.**
Sørlie et al (2001) *PNAS*. 98(19):10869-74.

# RNA-seq Overview

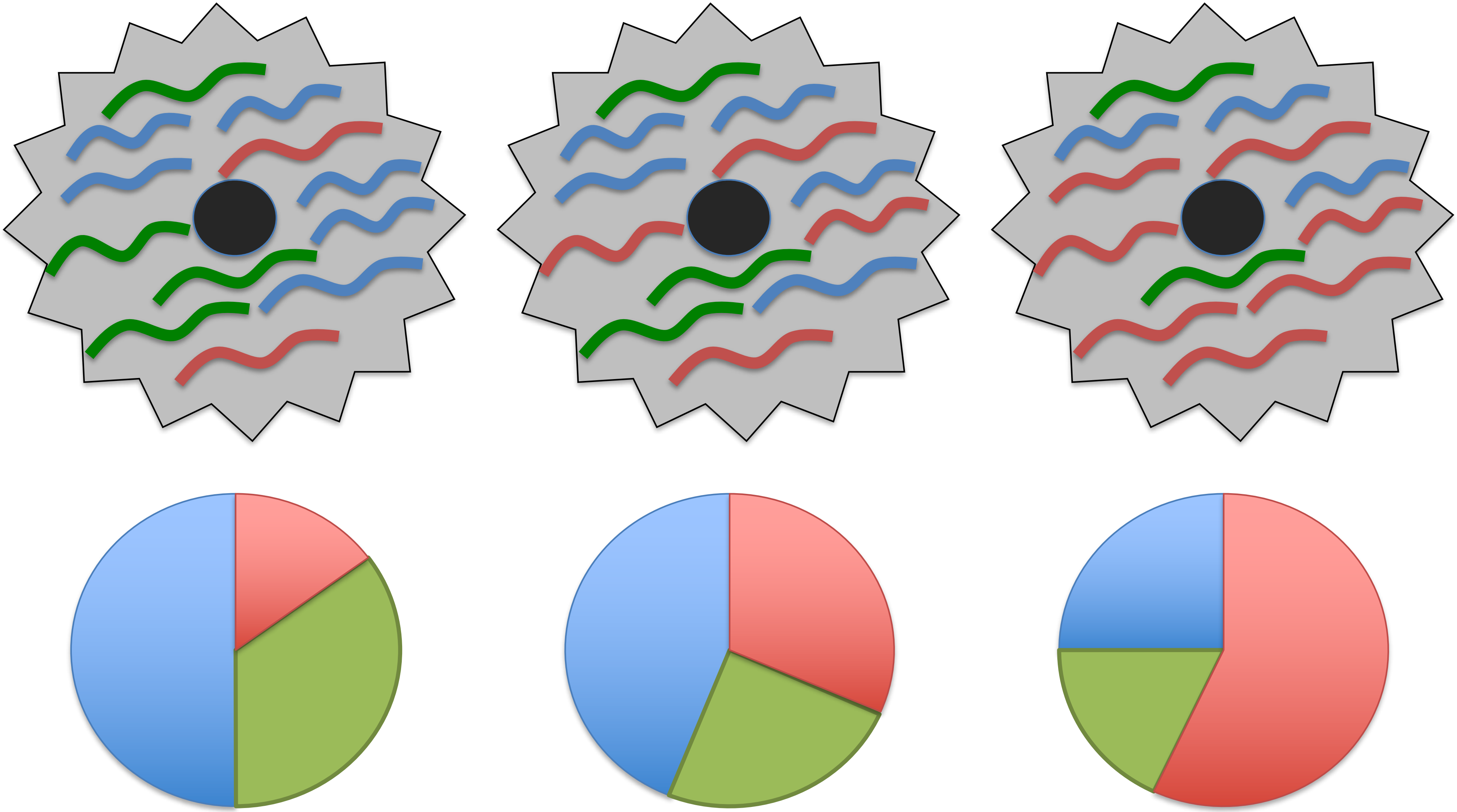


Sequencing

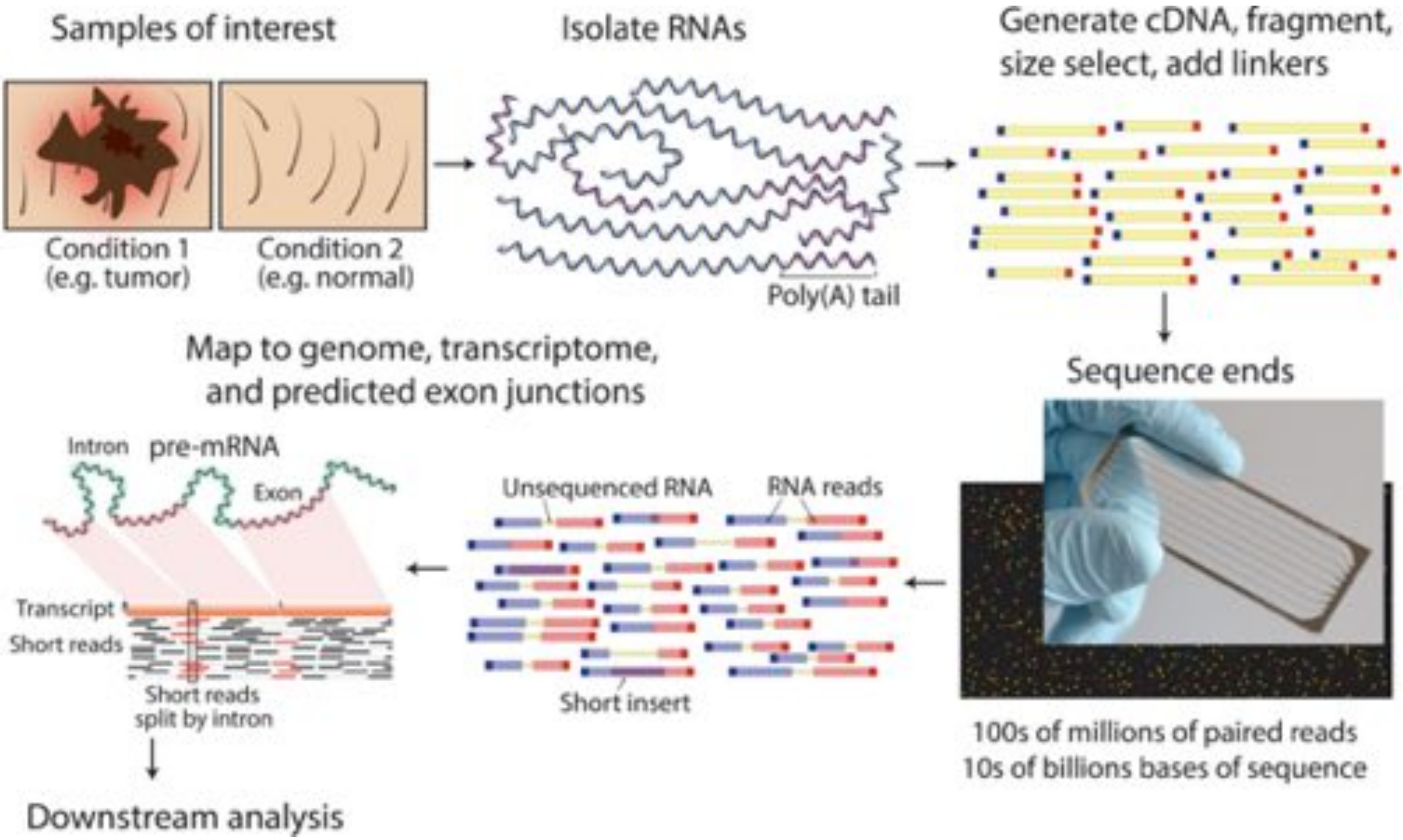
Mapping & Assembly

Quantification

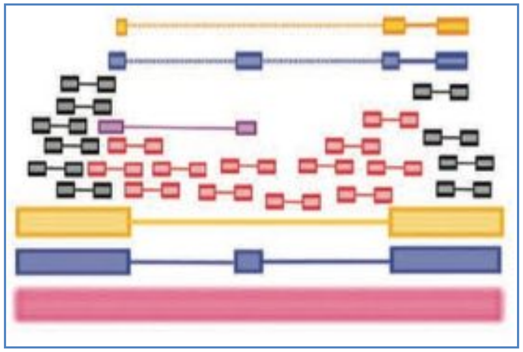
# RNA-seq Overview



# RNA-seq Overview



# RNA-seq Challenges



***Challenge 1: Eukaryotic genes are spliced***

# RNA-Seq Approaches

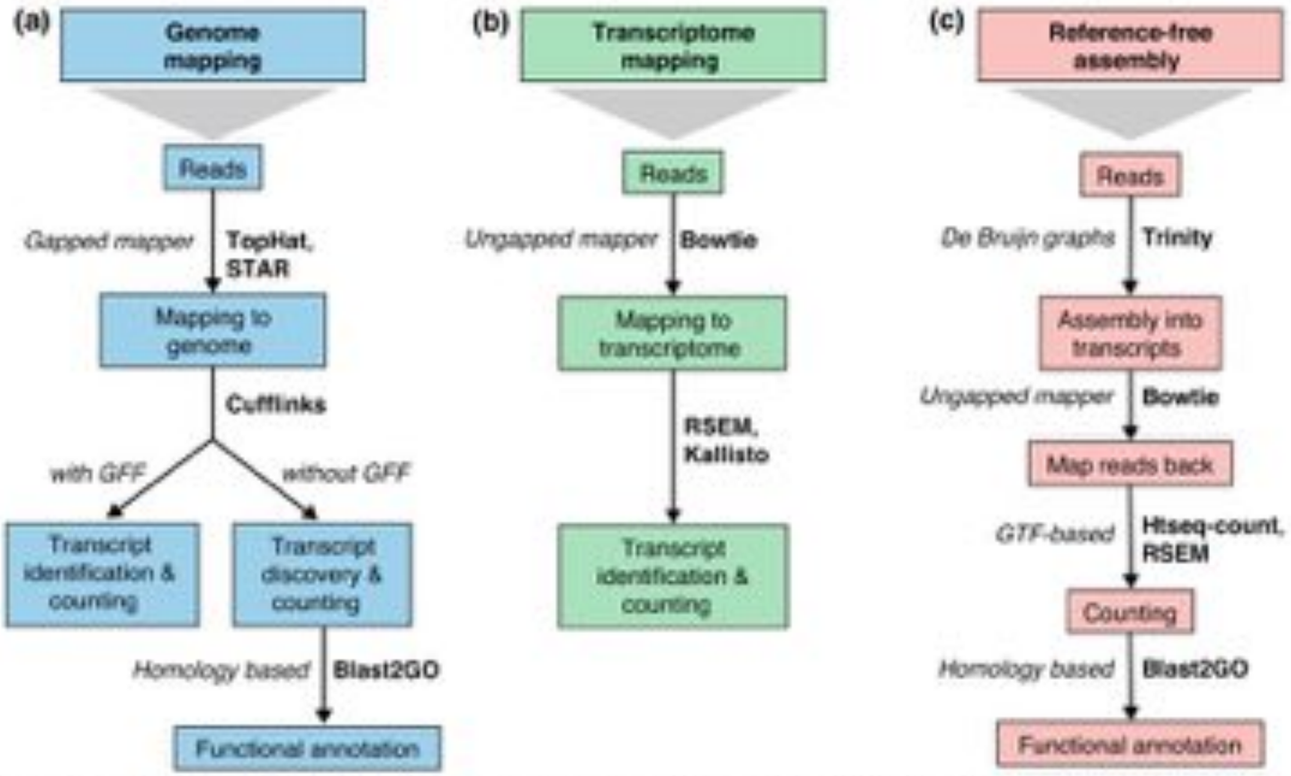


**Fig. 2** Read mapping and transcript identification strategies. Three basic strategies for regular RNA-seq analysis. **a** An annotated genome is available and reads are mapped to the genome with a gapped mapper. Next (novel) transcript discovery and quantification can proceed with or without an annotation file. Novel transcripts are then functionally annotated. **b** If no novel transcript discovery is needed, reads can be mapped to the reference transcriptome using an ungapped aligner. Transcript identification and quantification can occur simultaneously. **c** When no genome is available, reads need to be assembled first into contigs or transcripts. For quantification, reads are mapped back to the novel reference transcriptome and further analysis proceeds as in (**b**) followed by the functional annotation of the novel transcripts as in (**a**). Representative software that can be used at each analysis step are indicated in *bold text*. Abbreviations: *GFF* General Feature Format, *GTF* gene transfer format, *RSEM* RNA-Seq by Expectation Maximization

***A survey of best practices for RNA-seq data analysis***
Conesa et al  (2016) Genome Biology. doi 10.1186/s13059-016-0881-8

# RNA-Seq Approaches

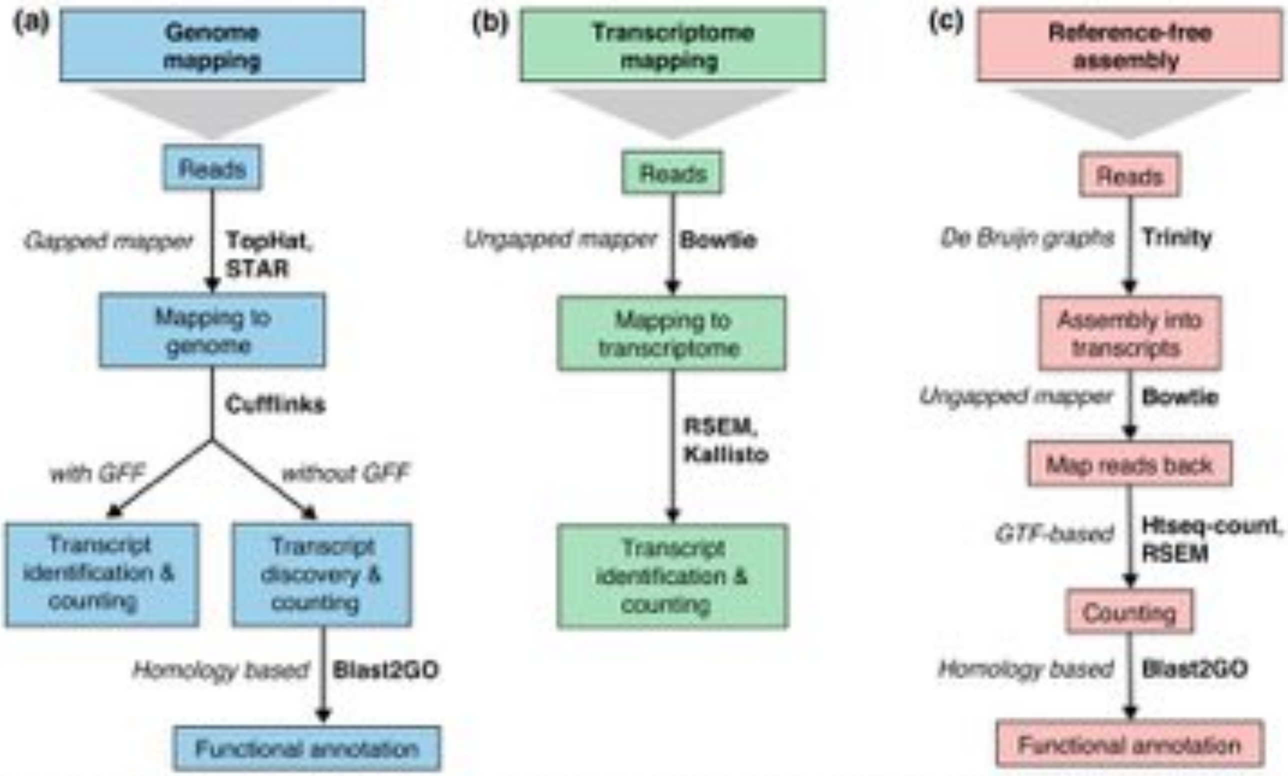


**Fig. 2** Read mapping and transcript identification strategies. Three basic strategies for regular RNA-seq analysis. **a** An annotated genome is available and re[illegible]vel) transcript discovery and quantification can proceed with or without an ann[illegible] no novel transcript discovery is needed, reads can be mapped to the reference transcriptome using an ungapped aligner. Transcript identification and quantification can occur simultaneously. **c** When no genome is available, reads n[illegible]o contigs or transcripts. For quantification, reads are mapped back to the novel reference transcriptome and further a[illegible]owed by the functional annotation of the novel transcripts as in (**a**). Representative software that can be used a[illegible]ated in *bold text*. Abbreviations: *GFF* General Feature Format, *GTF* gene transfer format, *RSEM* RNA-Seq by Expectation Maximization

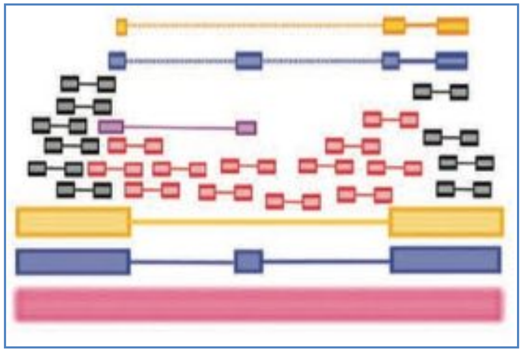
Which approach should we use?

It depends….

***A survey of best practices for RNA-seq data analysis***

Conesa et al  (2016) Genome Biology. doi 10.1186/s13059-016-0881-8

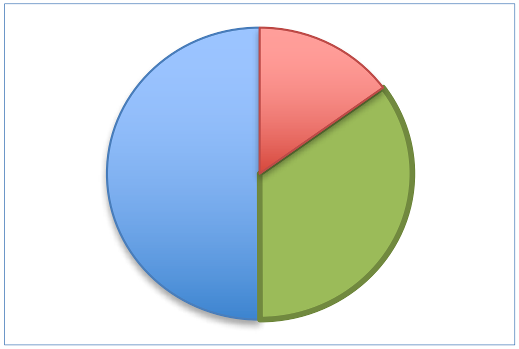
# RNA-seq Challenges



***Challenge 1: Eukaryotic genes are spliced***

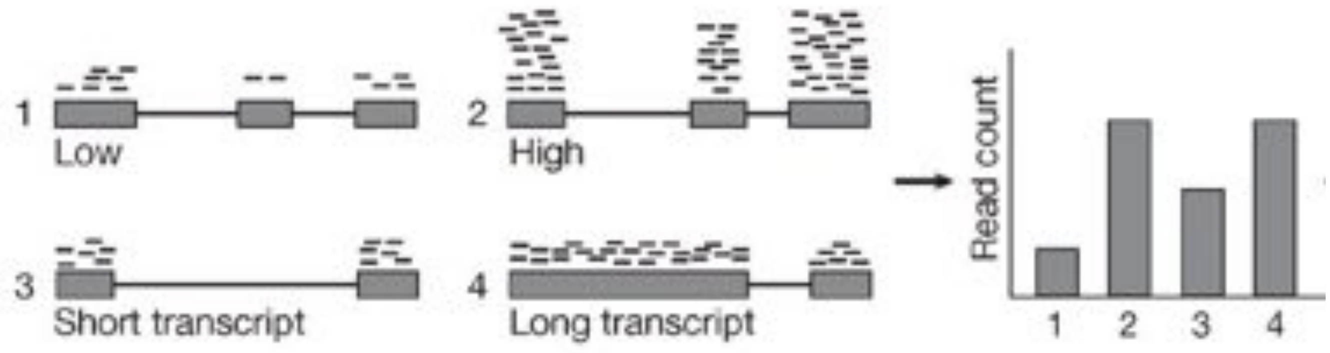
Solution: Use a spliced aligner, and assemble isoforms

***TopHat: discovering spliced junctions with RNA-Seq.***
Trapnell et al (2009) *Bioinformatics*. 25:0 1105-1111



***Challenge 2: Read Count != Transcript abundance***

# RPKM, FPKM, TPM

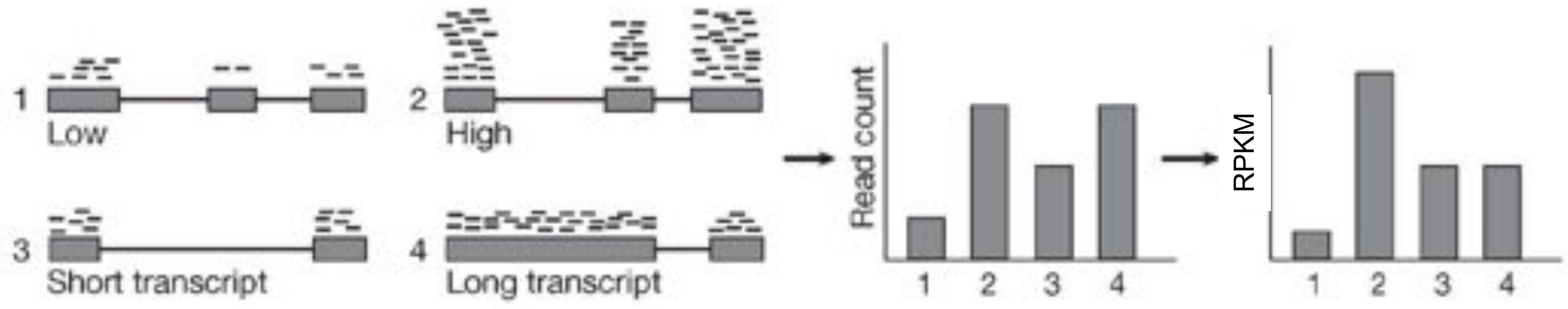


***Counting Reads that align to a gene DOESN'T work!***

- Overall Coverage: 1M reads in experiment 1 vs 10M reads in experiment 2
- Gene Length: gene 3 is 10kbp, gene 4 is 100kbp

***1. RPKM: Reads Per Kilobase of Exon Per Million Reads Mapped (Mortazavi et al, 2008)***

# RPKM, FPKM, TPM



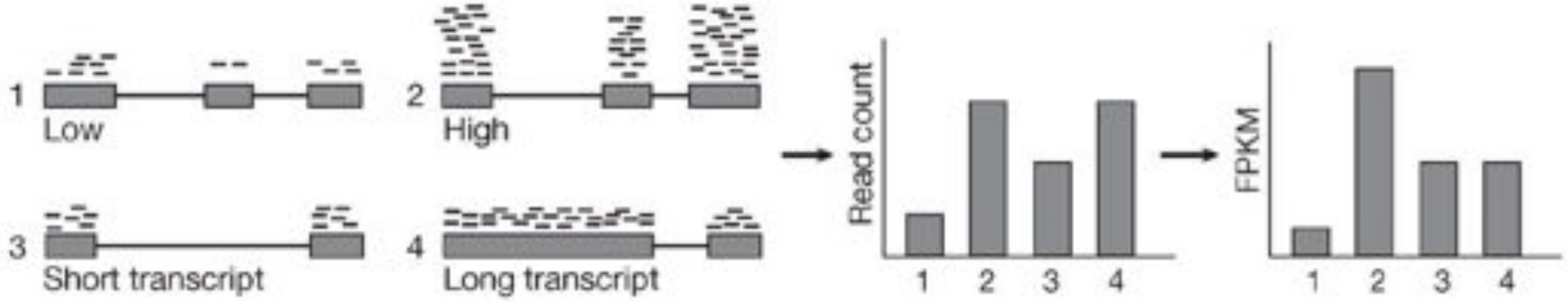
***Counting Reads that align to a gene DOESN'T work!***
- Overall Coverage: 1M reads in experiment 1 vs 10M reads in experiment 2
- Gene Length: gene 3 is 10kbp, gene 4 is 100kbp

***1. RPKM: Reads Per Kilobase of Exon Per Million Reads Mapped (Mortazavi et al, 2008)***

(Count reads aligned to gene) / (length of gene in kilobases) / (# millions of read mapped)

=> Wait a second, reads in a pair arent independent!

# RPKM, FPKM, TPM



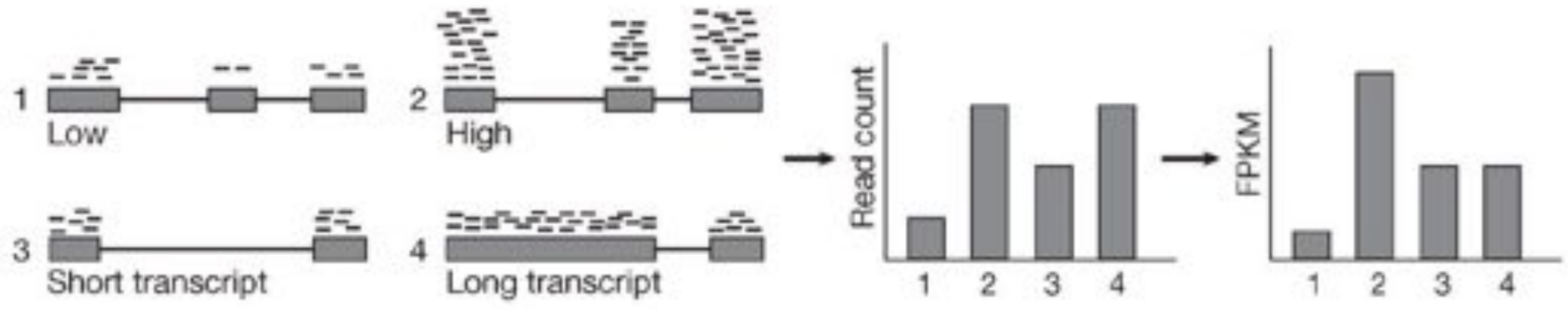
***Counting Reads that align to a gene DOESN'T work!***
- Overall Coverage: 1M reads in experiment 1 vs 10M reads in experiment 2
- Gene Length: gene 3 is 10kbp, gene 4 is 100kbp

***1. RPKM: Reads Per Kilobase of Exon Per Million Reads Mapped (Mortazavi et al, 2008)***
=> Wait a second, reads in a pair arent independent!

***2. FPKM: Fragments Per Kilobase of Exon Per Million Reads Mapped (Trapnell et al, 2010)***
⇒ Does a much better job with short exons & short genes by boosting coverage

⇒ Wait a second, FPKM depends on the average transcript length!

# RPKM, FPKM, TPM



***Counting Reads that align to a gene DOESN'T work!***
- Overall Coverage: 1M reads in experiment 1 vs 10M reads in experiment 2
- Gene Length: gene 3 is 10kbp, gene 4 is 100kbp

***1. RPKM: Reads Per Kilobase of Exon Per Million Reads Mapped (Mortazavi et al, 2008)***
=> Wait a second, reads in a pair arent independent!

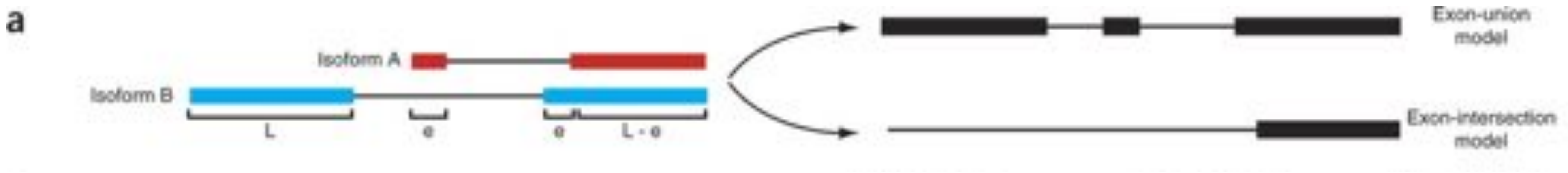
***2. FPKM: Fragments Per Kilobase of Exon Per Million Reads Mapped (Trapnell et al, 2010)***
=> Wait a second, FPKM depends on the average transcript length!

***3. TPM: Transcripts Per Million (Li et al, 2011)***
⇒ If you were to sequence one million full length transcripts, TPM is the number of transcripts you would have seen of type i, given the abundances of the other transcripts in your sample

=> Recommend you use TPM for all analysis, easy to compute given FPKM

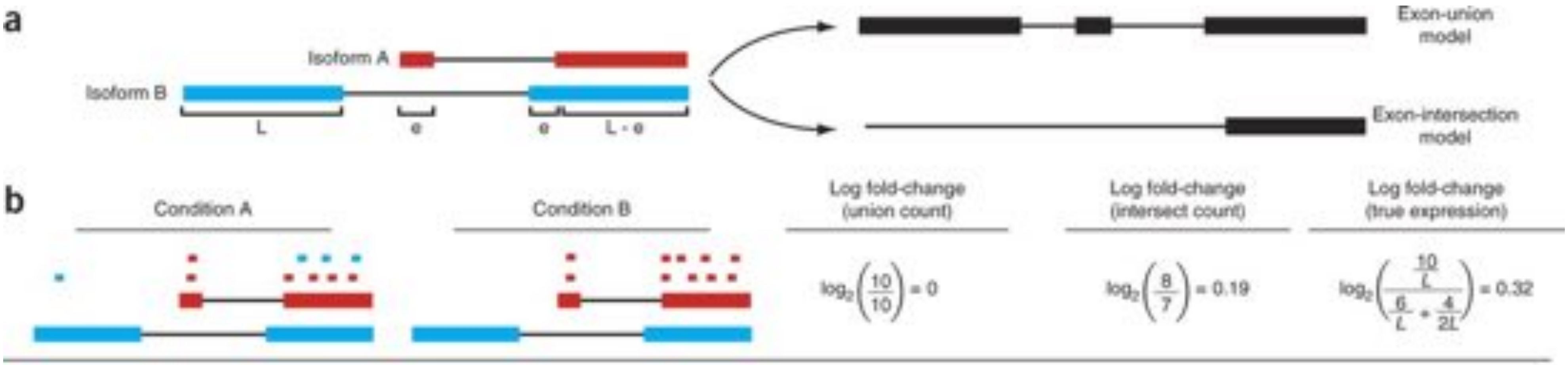
$$\mathrm{TPM}_i = \left( \frac{\mathrm{FPKM}_i}{\sum_j \mathrm{FPKM}_j} \right) \cdot 10^6$$

# Gene or Isoform Quantification?



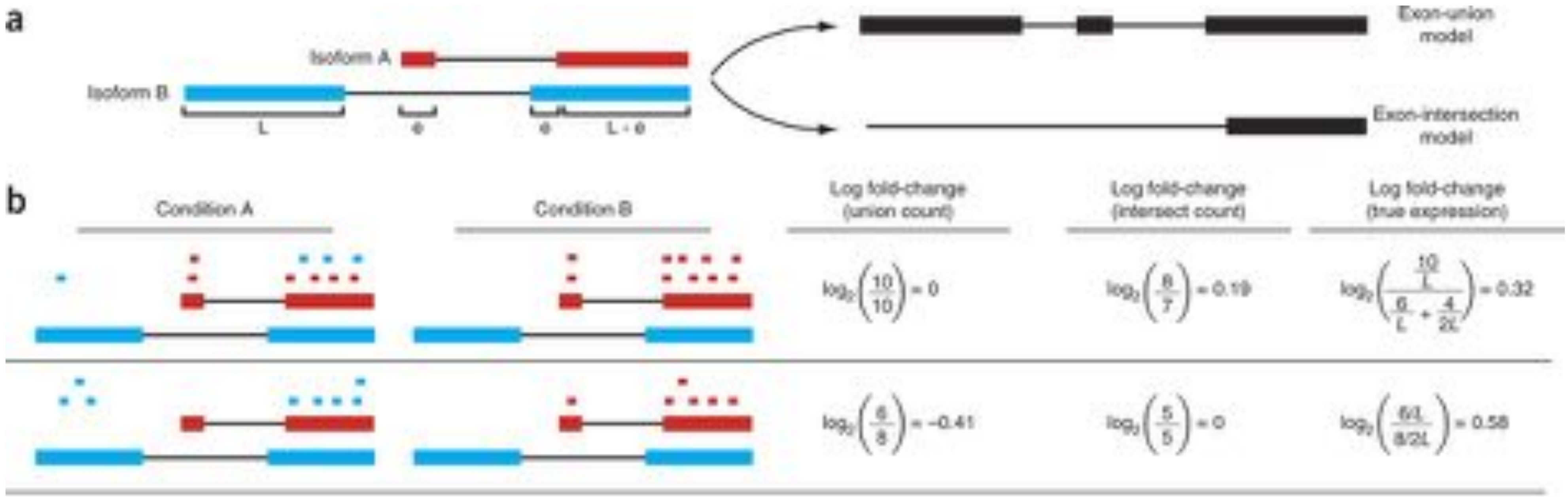
***Differential analysis of gene regulation at transcript resolution with RNA-seq***
Trapnell et al (2013) Nature Biotechnology 31, 46–53. doi:10.1038/nbt.2450

# Gene or Isoform Quantification?


a
Isoform A
Isoform B
L
e
e
L - e
Exon-union model
Exon-intersection model
b
Condition A
Condition B
Log fold-change (union count)
Log fold-change (intersect count)
Log fold-change (true expression)
$\log_2\left(\frac{10}{10}\right) = 0$
$\log_2\left(\frac{8}{7}\right) = 0.19$
$\log_2\left(\frac{\frac{10}{L}}{\frac{6}{L} + \frac{4}{2L}}\right) = 0.32$

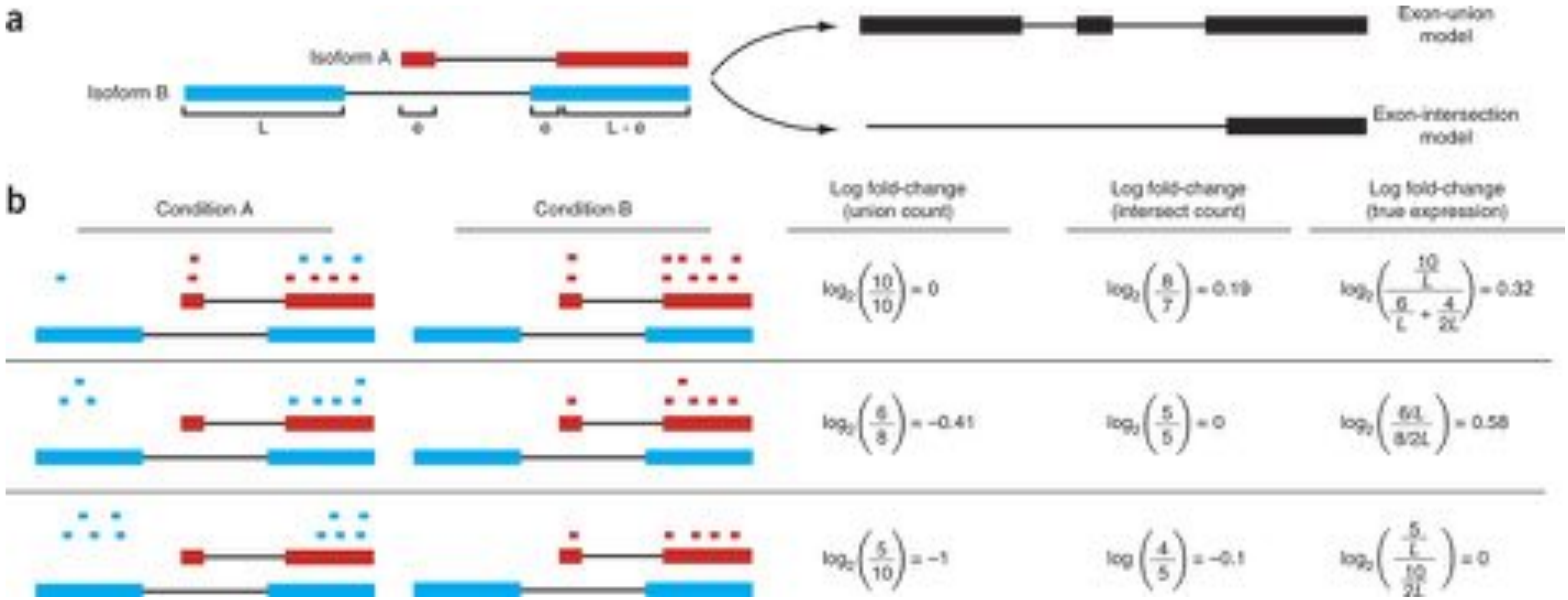
***Differential analysis of gene regulation at transcript resolution with RNA-seq***
Trapnell et al (2013) Nature Biotechnology 31, 46–53. doi:10.1038/nbt.2450

# Gene or Isoform Quantification?


a
Isoform A
Isoform B
L
e
e
L - e
Exon-union model
Exon-intersection model
b
Condition A
Condition B
Log fold-change (union count)
Log fold-change (intersect count)
Log fold-change (true expression)
$\log_2\left(\frac{10}{10}\right) = 0$
$\log_2\left(\frac{8}{7}\right) = 0.19$
$\log_2\left(\frac{\frac{10}{L}}{\frac{6}{L}+\frac{4}{2L}}\right) = 0.32$
$\log_2\left(\frac{6}{8}\right) = -0.41$
$\log_2\left(\frac{5}{5}\right) = 0$
$\log_2\left(\frac{6/L}{8/2L}\right) = 0.58$

***Differential analysis of gene regulation at transcript resolution with RNA-seq***
Trapnell et al (2013) Nature Biotechnology 31, 46–53. doi:10.1038/nbt.2450

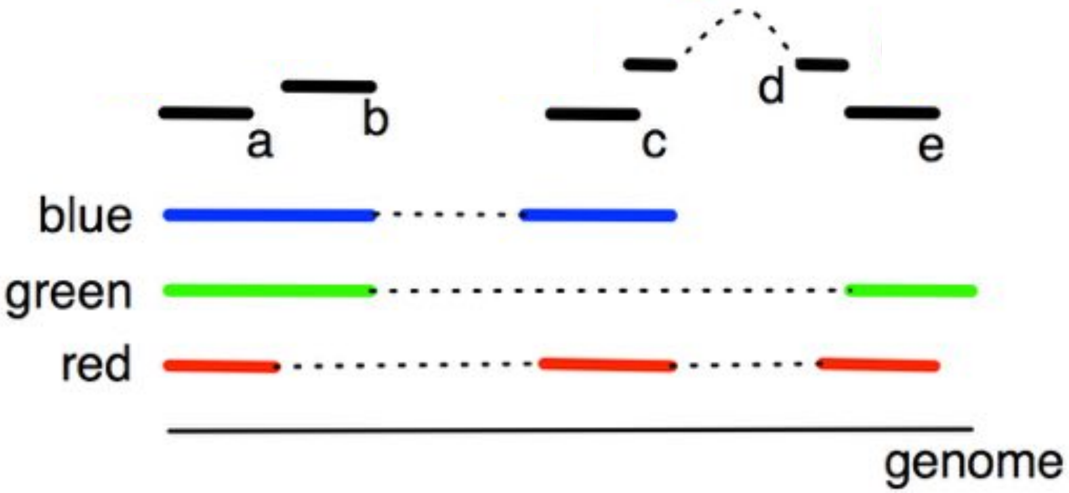
# Gene or Isoform Quantification?


a
Isoform A
Isoform B
L
e
e
L - e
Exon-union model
Exon-intersection model
b
Condition A
Condition B
Log fold-change (union count)
Log fold-change (intersect count)
Log fold-change (true expression)
$\log_2\left(\frac{10}{10}\right) = 0$
$\log_2\left(\frac{8}{7}\right) = 0.19$
$\log_2\left(\frac{\frac{10}{L}}{\frac{6}{L}+\frac{4}{2L}}\right) = 0.32$
$\log_2\left(\frac{6}{8}\right) = -0.41$
$\log_2\left(\frac{5}{5}\right) = 0$
$\log_2\left(\frac{6/L}{8/2L}\right) = 0.58$
$\log_2\left(\frac{5}{10}\right) = -1$
$\log\left(\frac{4}{5}\right) = -0.1$
$\log_2\left(\frac{\frac{5}{L}}{\frac{10}{2L}}\right) = 0$

***Key point : The length of the actual molecule from which the fragments derive is crucially important to obtaining accurate abundance estimates.***

***Differential analysis of gene regulation at transcript resolution with RNA-seq***
Trapnell et al (2013) Nature Biotechnology 31, 46–53. doi:10.1038/nbt.2450

# Multi-mapping? Isoform ambiguity? Expectation Maximization to the Rescue



The gene has three isoforms (red, green, blue) of the same length.
Our initial expectation is all 3 isoforms are equally expressed

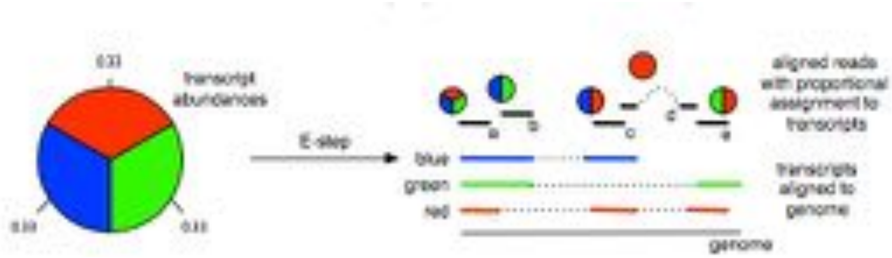
There are five reads (a,b,c,d,e) mapping to the gene.

- Read a maps to all three isoforms
- Read d only to red
- Reads b,c,e map to each of the three pairs of isoforms.

What is the most likely expression level of each isoform?

***Models for transcript quantification from RNA-seq***
Pachter, L (2011) arXiv. 1104.3889 [q-bio.GN]

# Multi-mapping? Isoform ambiguity?
# Expectation Maximization to the Rescue


0.33
transcript abundances
0.33
0.33
E-step
blue
green
red
a
b
c
d
e
aligned reads with proportional assignment to transcripts
transcripts aligned to genome
genome

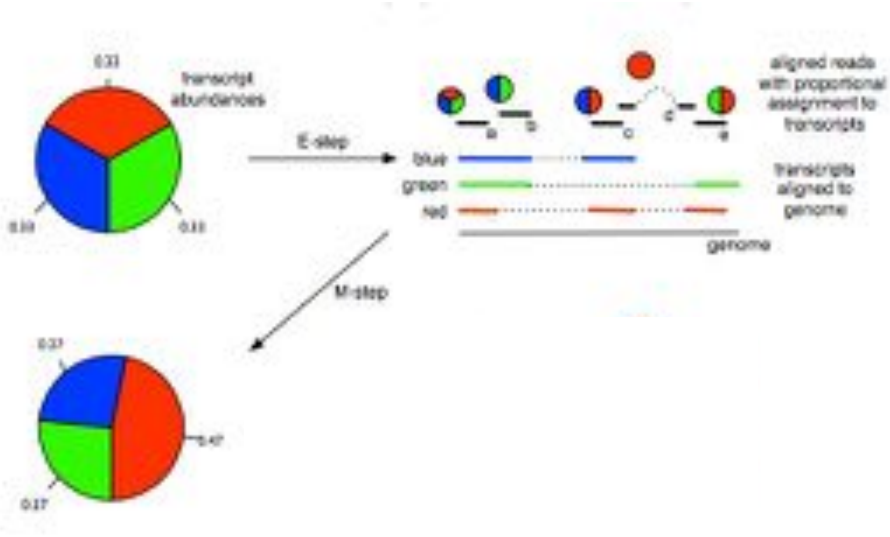
The gene has three isoforms (red, green, blue) of the same length. Initially every isoform is assigned the same abundance (red=1/3, green=1/3, blue=1/3)

There are five reads (a,b,c,d,e) mapping to the gene. Read a maps to all three isoforms, read d only to red, and the other three (reads b,c,e) to each of the three pairs of isoforms.

During the expectation (E) step reads are proportionately assigned to transcripts according to the (current) isoform abundances (RGB): a=(.33,.33,.33), b=(0,.5,.5), c=(.5,.5), d=(1,0,0), e=(.5,.5,0)

***Models for transcript quantification from RNA-seq***
Pachter, L (2011) arXiv. 1104.3889 [q-bio.GN]

# Multi-mapping? Isoform ambiguity?
# Expectation Maximization to the Rescue



0.33
transcript abundances
0.33
0.33
E-step
aligned reads with proportional assignment to transcripts
a
b
c
d
e
blue
green
red
transcripts aligned to genome
genome
M-step
0.27
0.47
0.27

The gene has three isoforms (red, green, blue) of the same length. Initially every isoform is assigned the same abundance (red=1/3, green=1/3, blue=1/3)

There are five reads (a,b,c,d,e) mapping to the gene. Read a maps to all three isoforms, read d only to red, and the other three (reads b,c,e) to each of the three pairs of isoforms.

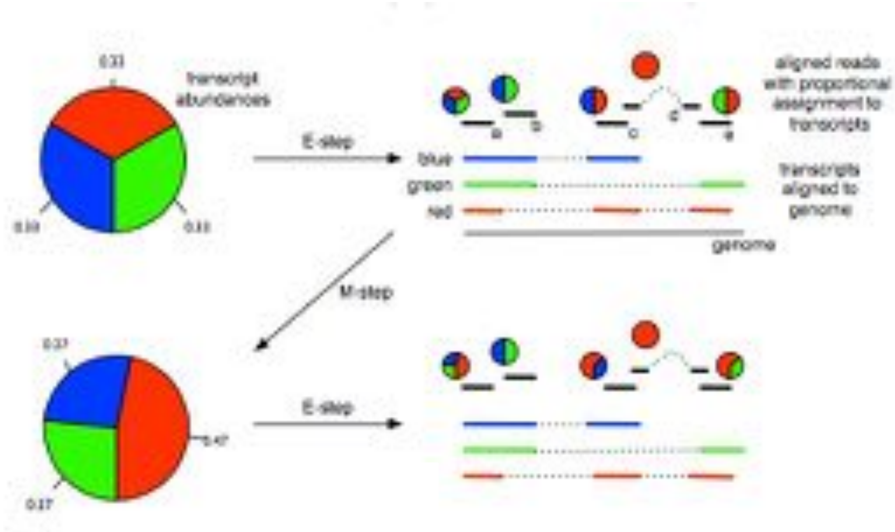
During the expectation (E) step reads are proportionately assigned to transcripts according to the (current) isoform abundances (RGB): a=(.33,.33,.33), b=(0,.5,.5), c=(.5,.5), d=(1,0,0), e=(.5,.5,0)

Next, during the maximization (M) step isoform abundances are recalculated from the proportionately assigned read counts:

red: 0.47 = (0.33 + 0.5 + 1 + 0.5)/(2.33 + 1.33 + 1.33)
blue: 0.27 = (0.33 + 0.5 + 0.5)/(2.33 + 1.33 + 1.33)
green: 0.27 = (0.33 + 0.5 + 0.5)/(2.33 + 1.33 + 1.33)

***Models for transcript quantification from RNA-seq***
Pachter, L (2011) arXiv. 1104.3889 [q-bio.GN]

# Multi-mapping? Isoform ambiguity?
# Expectation Maximization to the Rescue


0.33
transcript abundances
0.33
0.33
E-step
aligned reads with proportional assignment to transcripts
a
b
c
d
e
blue
green
red
transcripts aligned to genome
genome
M-step
0.27
0.47
0.27
E-step

The gene has three isoforms (red, green, blue) of the same length. Initially every isoform is assigned the same abundance (red=1/3, green=1/3, blue=1/3)

There are five reads (a,b,c,d,e) mapping to the gene. Read a maps to all three isoforms, read d only to red, and the other three (reads b,c,e) to each of the three pairs of isoforms.

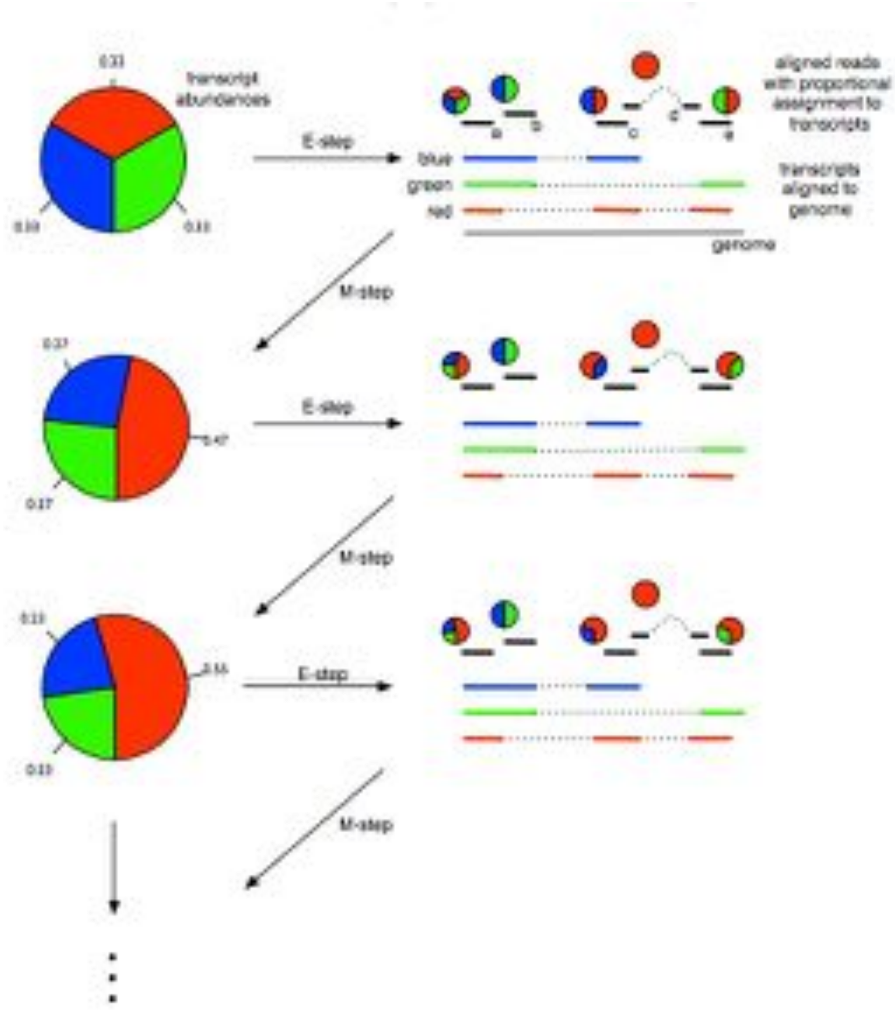
During the expectation (E) step reads are proportionately assigned to transcripts according to the (current) isoform abundances (RGB): a=(.33,.33,.33), b=(0,.5,.5), c=(.5,.5), d=(1,0,0), e=(.5,.5,0)

Next, during the maximization (M) step isoform abundances are recalculated from the proportionately assigned read counts:
red: 0.47 = (0.33 + 0.5 + 1 + 0.5)/(2.33 + 1.33 + 1.33)
blue: 0.27 = (0.33 + 0.5 + 0.5)/(2.33 + 1.33 + 1.33)
green: 0.27 = (0.33 + 0.5 + 0.5)/(2.33 + 1.33 + 1.33)

Repeat until convergence!

***Models for transcript quantification from RNA-seq***
Pachter, L (2011) arXiv. 1104.3889 [q-bio.GN]

# Multi-mapping? Isoform ambiguity? Expectation Maximization to the Rescue


0.33
transcript abundances
0.33
0.33
E-step
M-step
aligned reads with proportional assignment to transcripts
a
b
c
d
e
blue
green
red
transcripts aligned to genome
genome
0.27
0.47
0.27

The gene has three isoforms (red, green, blue) of the same length. Initially every isoform is assigned the same abundance (red=1/3, green=1/3, blue=1/3)

There are five reads (a,b,c,d,e) mapping to the gene. Read a maps to all three isoforms, read d only to red, and the other three (reads b,c,e) to each of the three pairs of isoforms.

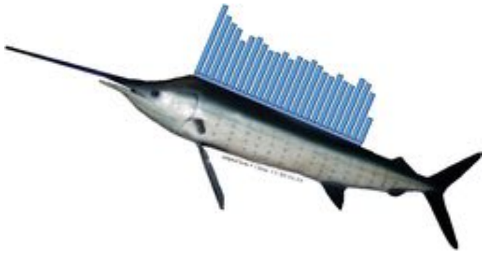
During the expectation (E) step reads are proportionately assigned to transcripts according to the (current) isoform abundances (RGB): a=(.33,.33,.33), b=(0,.5,.5), c=(.5,.5), d=(1,0,0), e=(.5,.5,0)

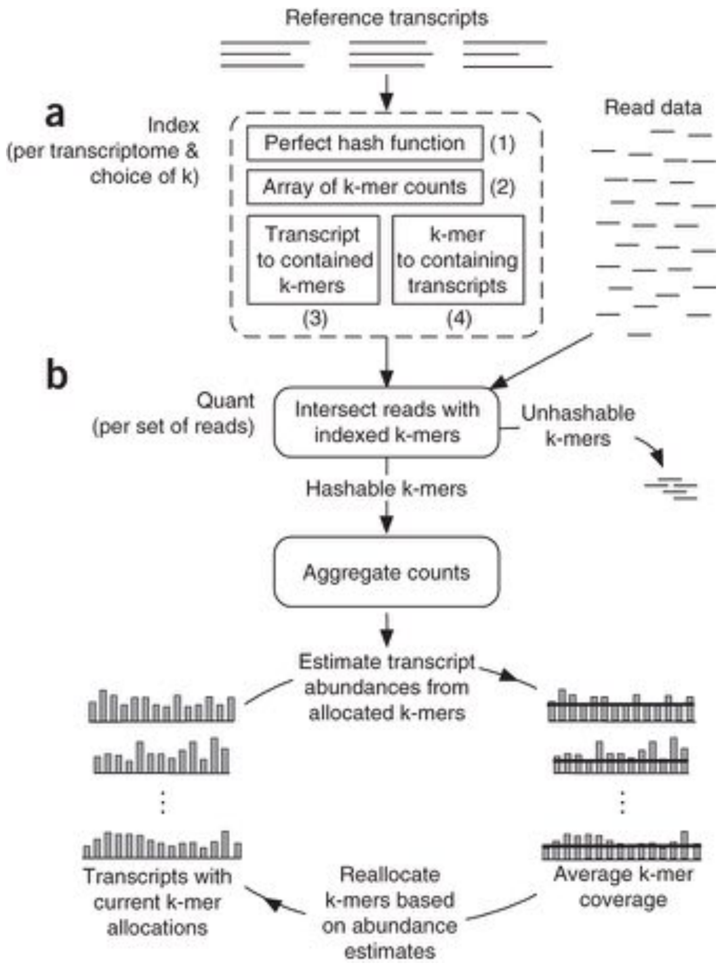
Next, during the maximization (M) step isoform abundances are recalculated from the proportionately assigned read counts:
red: 0.47 = (0.33 + 0.5 + 1 + 0.5)/(2.33 + 1.33 + 1.33)
blue: 0.27 = (0.33 + 0.5 + 0.5)/(2.33 + 1.33 + 1.33)
green: 0.27 = (0.33 + 0.5 + 0.5)/(2.33 + 1.33 + 1.33)

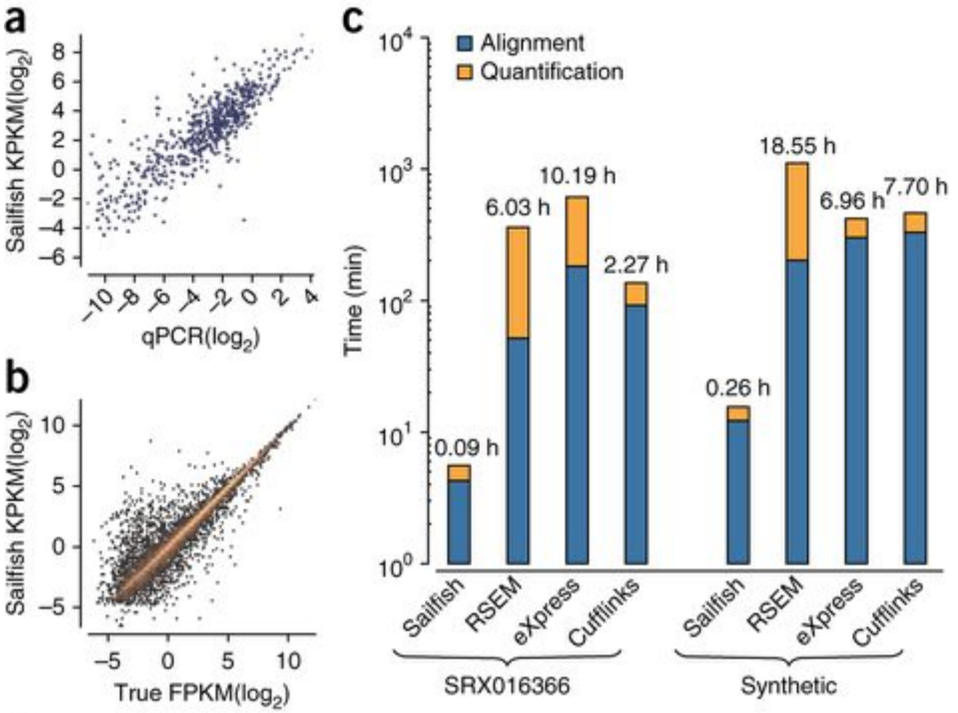
Repeat until convergence!

***Models for transcript quantification from RNA-seq***
Pachter, L (2011) arXiv. 1104.3889 [q-bio.GN]

# Sailfish: Fast & Accurate RNA-seq Quantification







d

| | Human brain tissue | | | | Synthetic | | | |
|---|---|---|---|---|---|---|---|---|
| | Sailfish | RSEM | eXpress | Cufflinks | Sailfish | RSEM | eXpress | Cufflinks |
| Pearson | 0.85 | 0.82 | 0.85 | 0.85 | 0.96 | 0.96 | 0.95 | 0.94 |
| Spearman | 0.84 | 0.80 | 0.85 | 0.85 | 0.76 | 0.77 | 0.77 | 0.75 |
| RMSE | – | – | – | – | 6.06 | 8.90 | 8.83 | 10.05 |
| medPE | – | – | – | – | 6.50 | 12.48 | 14.06 | 12.42 |

***Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms***
Patro et al (2014) Nature Biotechnology 32, 462–464 doi:10.1038/nbt.2862

# Annotation Summary

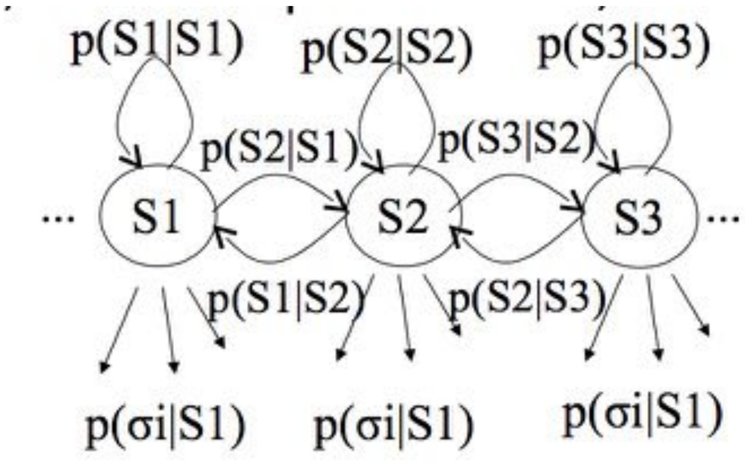
- Three major approaches to annotate a genome
  1. Alignment:
     - Does this sequence align to any other sequences of known function?
     - Great for projecting knowledge from one species to another
  2. Prediction:
     - Does this sequence statistically resemble other known sequences?
     - Potentially most flexible but dependent on good training data
  3. Experimental:
     - Lets test to see if it is transcribed/methylated/bound/etc
     - Strongest but expensive and context dependent

- Many great resources available
  – Learn to love the literature and the databases
  – Standard formats let you rapidly query and cross reference
  – Google is your number one resource ☺

# Machine Learning Primer:

# Hidden Markov Models

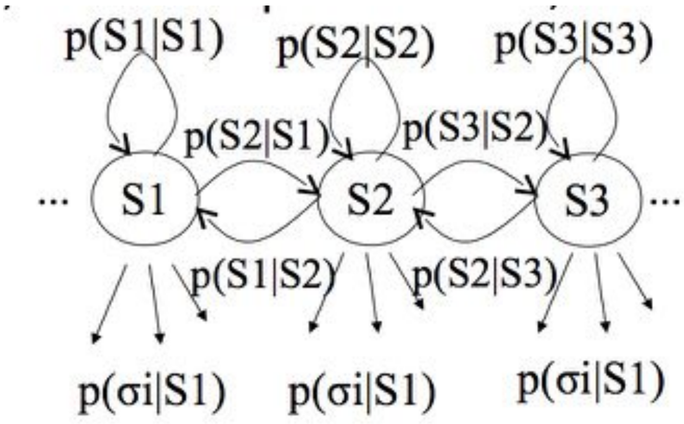
# What is an HMM?



- Dynamic Bayesian Network
  - A set of states
    - {Fair, Biased} for coin tossing
    - {Gene, Not Gene} for Bacterial Gene
    - {Intergenic, Exon, Intron} for Eukaryotic Gene
    - {Modern, Neanderthal} for Ancestry
  - A set of emission characters
    - E={H,T} for coin tossing
    - E={1,2,3,4,5,6} for dice tossing
    - E={A,C,G,T} for DNA
  - State-specific emission probabilities
    - P(H | Fair) = .5, P(T | Fair) = .5, P(H | Biased) = .9, P(T | Biased) = .1
    - P(A | Gene) = .9, P(A | Not Gene) = .1 …
  - A probability of taking a transition
    - P($s_i$=Fair|$s_{i-1}$=Fair) = .9, P($s_i$=Bias|$s_{i-1}$ = Fair) .1
    - P($s_i$=Exon | $s_{i-1}$=Intergenic), …

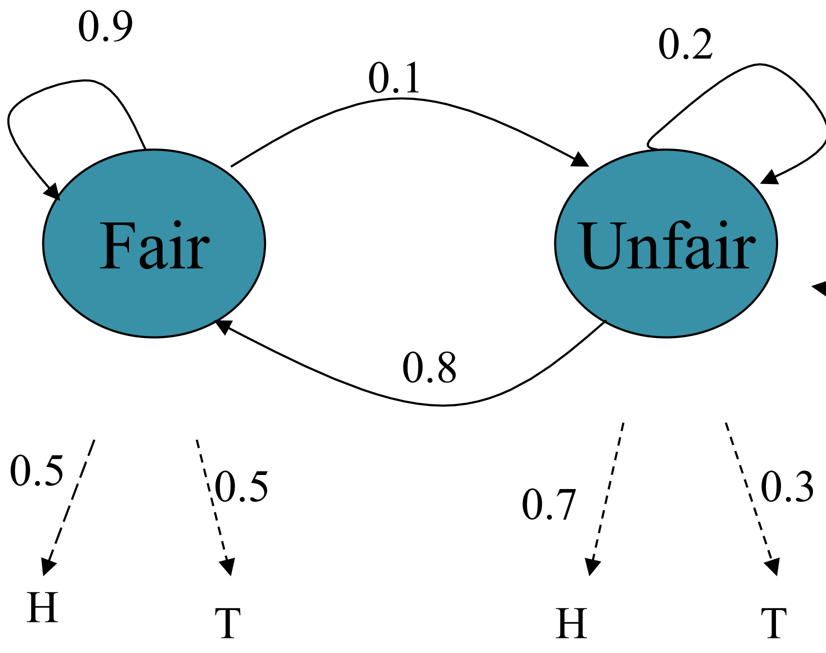
# Why Hidden?

- Observers can see the emitted symbols of an HMM (i.e., nucleotides) but have no ability to know which state the HMM is currently in (exon/intron/intergenic/etc).
  - But we can ***infer*** the most likely hidden states of an HMM based on the given sequence of emitted symbols.



AAAGCATGCATTTAACGTGAGCACAATAGATTACA

# HMM Example - Casino Coin



State transition probs.

States

Symbol emission probs.

Observation Symbols

**HTHHTTHHHTHTHTHHTHHHHHHTHTHH**

Observation Sequence

**FFFFFFUUUFFFFFFUUUUUUUFFFFFF**

State Sequence

**Motivation:** Given a sequence of H & Ts, can you tell at what times the casino cheated?

Slide credit: Fatih Gelgi, Arizona State U.

# Three classic HMM problems

1. **Evaluation**: given a model and an output sequence, what is the probability that the model generated that output?

2. **Decoding**: given a model and an output sequence, what is the most likely state sequence through the model that generated the output?

3. **Learning**: given a model and a set of observed sequences, how do we set the model's parameters so that it has a high probability of generating those sequences?

# Three classic HMM problems

1. **Evaluation**: given a model and an output sequence, what is the probability that the model generated that output?

- To answer this, we consider all possible paths through the model

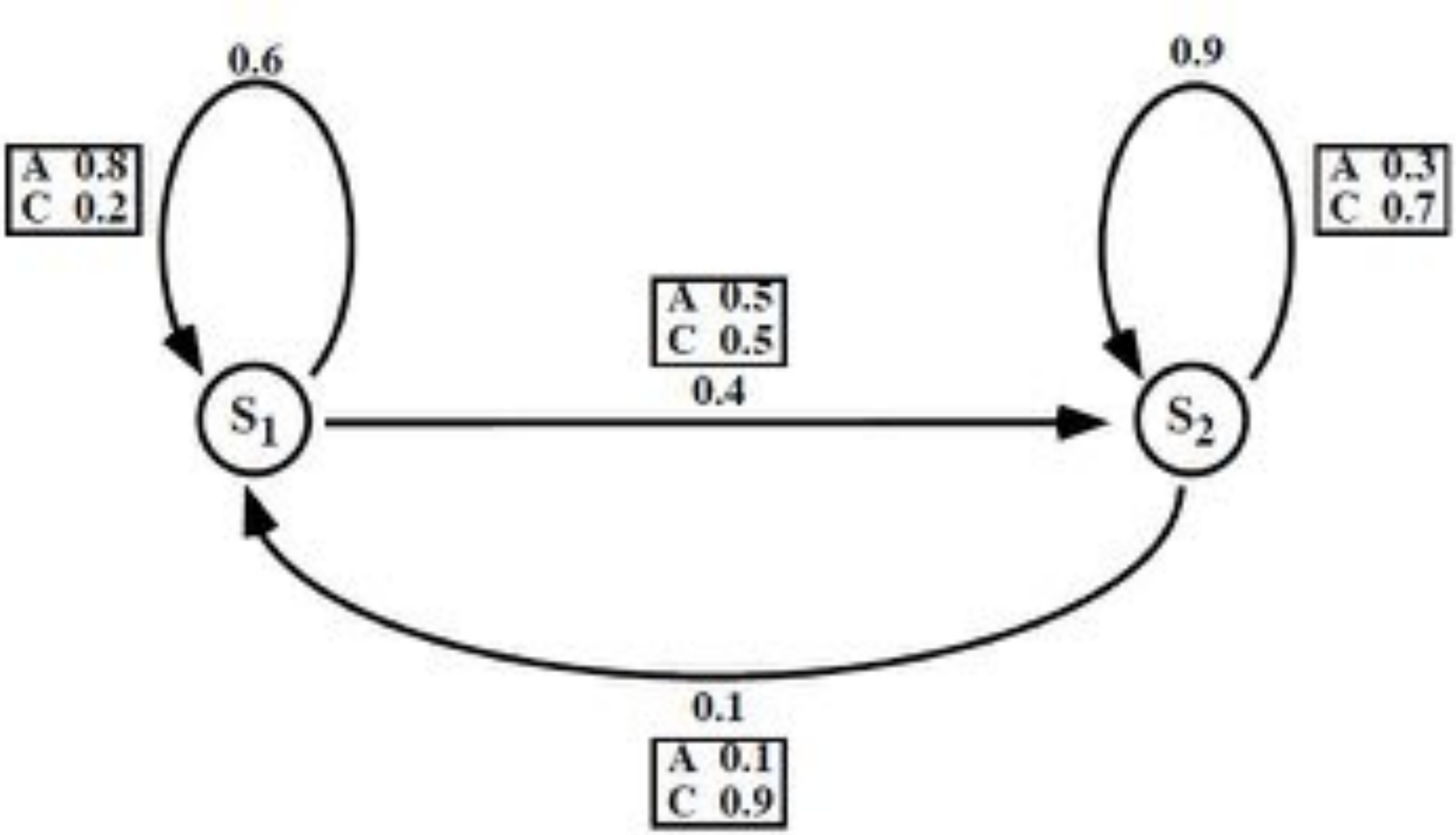
- Example: we might have a set of HMMs representing protein families -> pick the model with the best score

# Solving the Evaluation problem: The Forward algorithm

- To solve the Evaluation problem (probability that the model generated the sequence), we use the HMM and the data to build a *trellis*

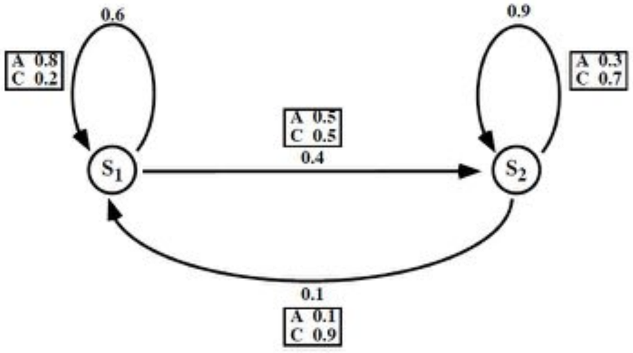
- Filling in the trellis will give tell us the probability that the HMM generated the data by finding all possible paths that could do it
  - Especially useful to evaluate from which models, a given sequence is most likely to have originated

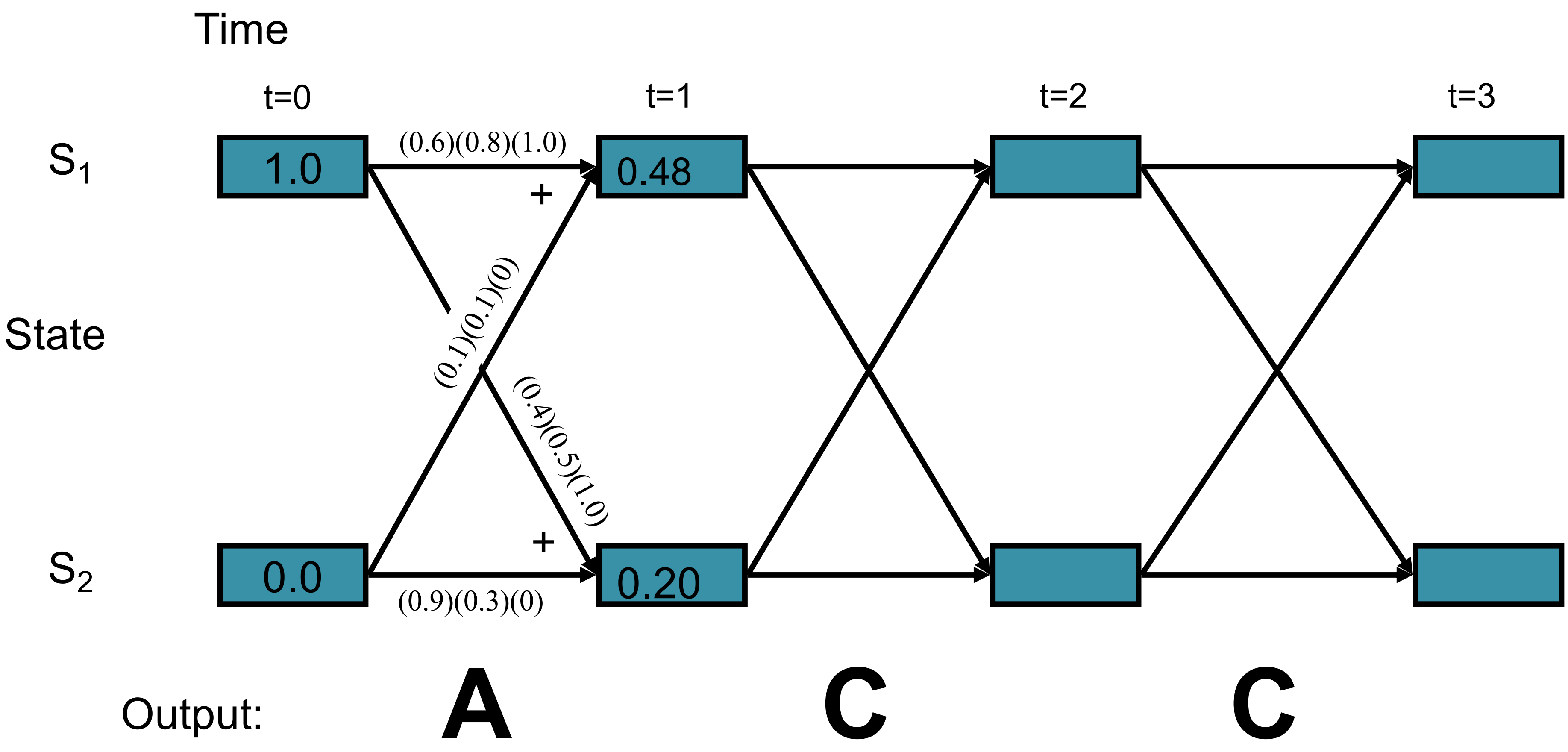
# Our sample HMM



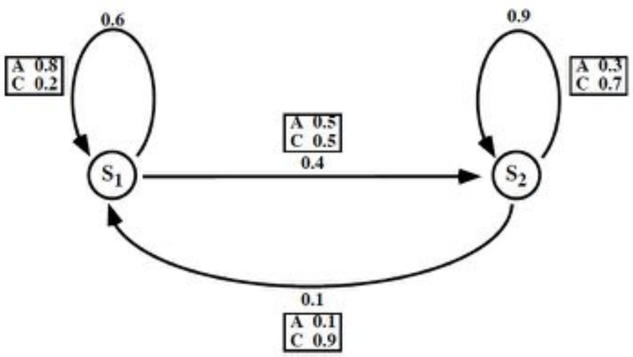
Let $S_1$ be initial state, $S_2$ be final state

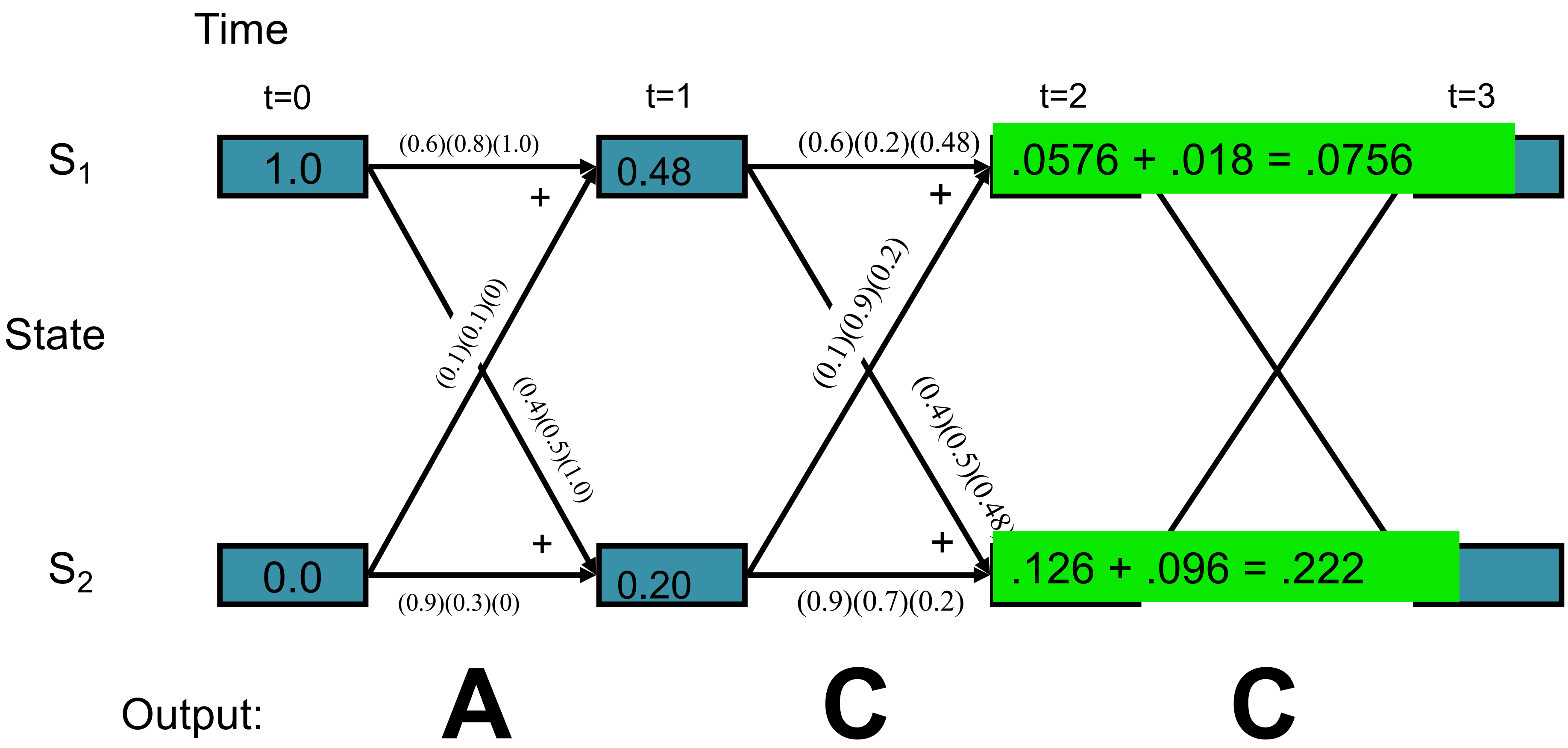
# A trellis for the Forward Algorithm



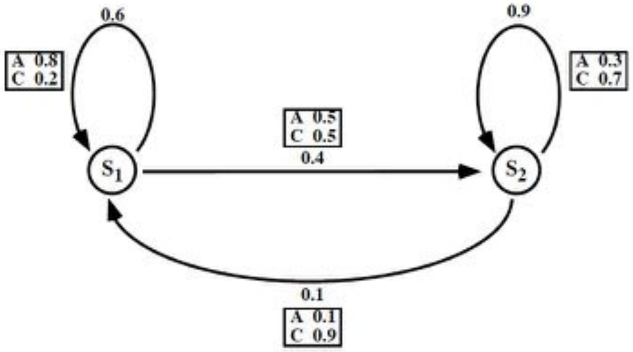


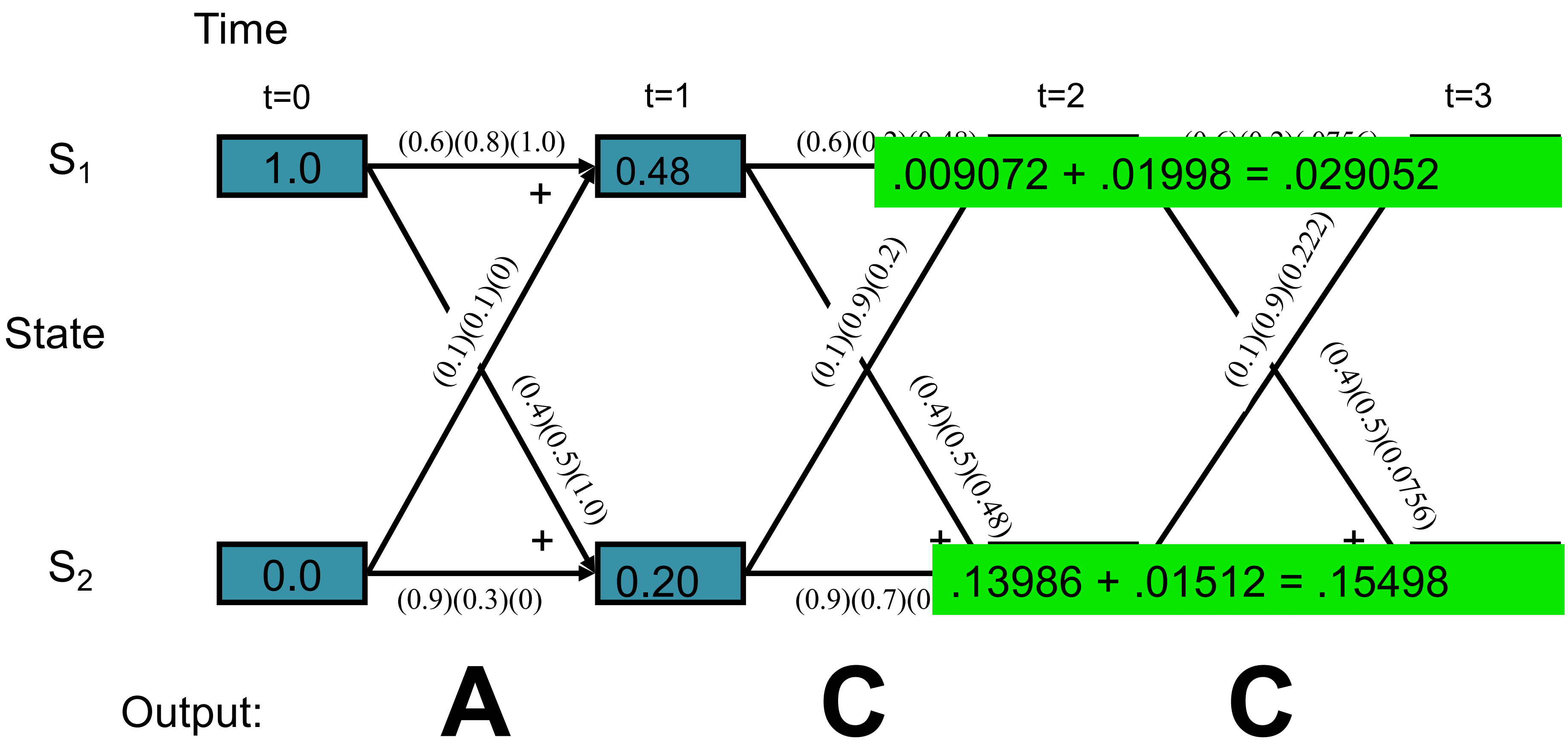
# A trellis for the Forward Algorithm





# A trellis for the Forward Algorithm

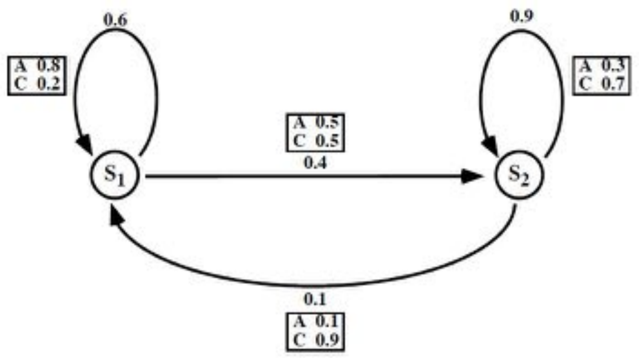


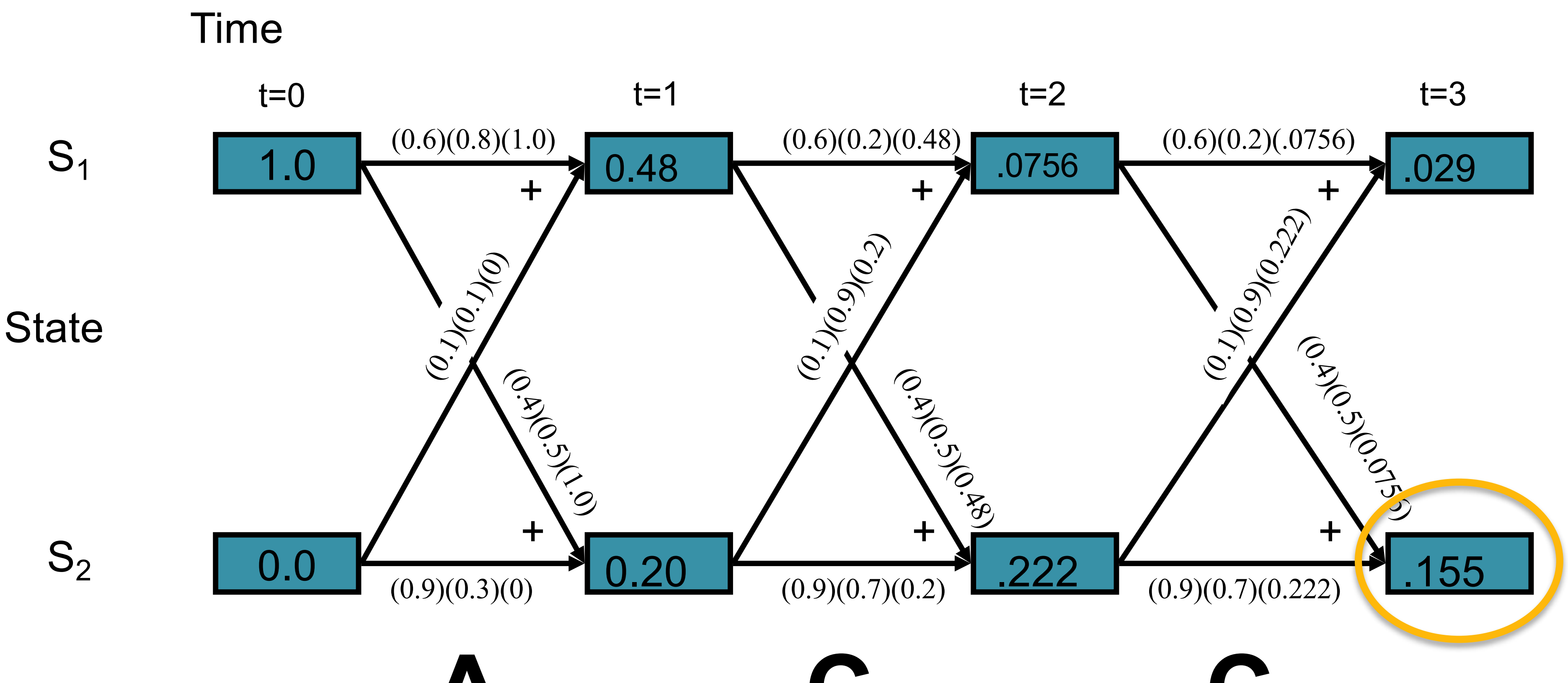


.009072 + .01998 = .029052

.13986 + .01512 = .15498

# A trellis for the Forward Algorithm





S2 is final state➔ 15.5% probability of this sequence given this model was used

# Probability of the model

- The Forward algorithm computes *P(y|M)*

- If we are comparing two or more models, we want the likelihood that each model generated the data: *P(M|y)*
  - Use Bayes' law: $$P(M \mid y) = \frac{P(y \mid M)P(M)}{P(y)}$$
  - Since P(y) is constant for a given input, we just need to maximize *P(y|M)P(M)*

# Three classic HMM problems

2. **Decoding**: given a model and an output sequence, what is the most likely state sequence through the model that generated the output?

- A solution to this problem gives us a way to match up an observed sequence and the states in the model.

AAAGCATGCATTTAACGAGAGCACAAGGGCTCTAATGCCG

The sequence of states is an annotation of the generated string – each nucleotide is generated in intergenic, start/stop, coding state

# Three classic HMM problems

2. **Decoding**: given a model and an output sequence, what is the most likely state sequence through the model that generated the output?

- A solution to this problem gives us a way to match up an observed sequence and the states in the model.

AAAGC ATG CAT TTA ACG AGA GCA CAA GGG CTC TAA TGCCG

The sequence of states is an annotation of the generated string – each nucleotide is generated in intergenic, start/stop, coding state

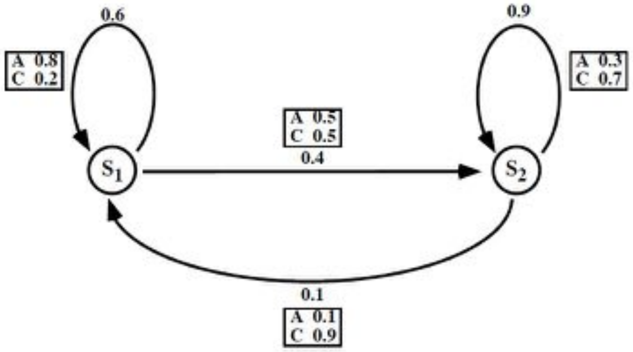
# Solving the Decoding Problem: The Viterbi algorithm

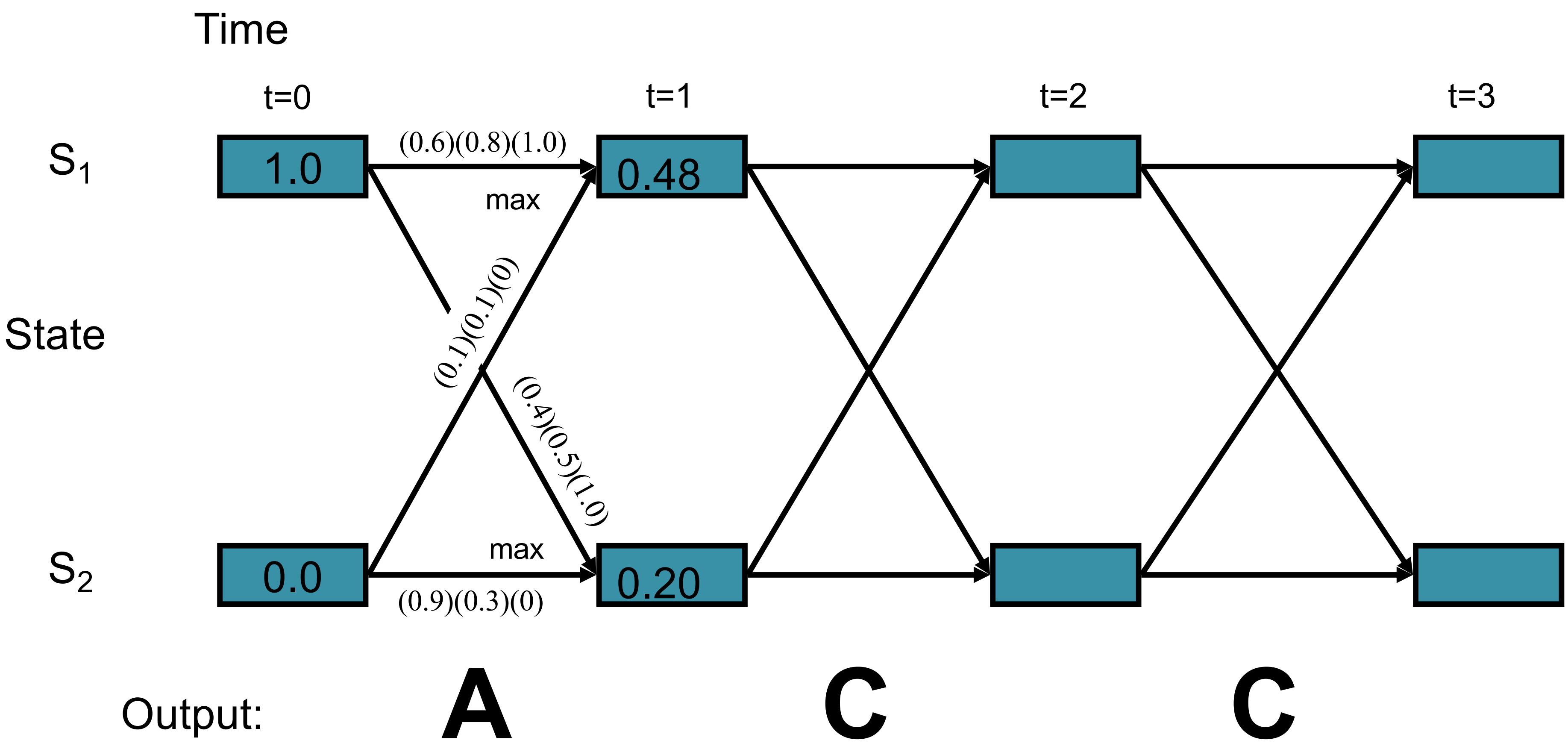
- To solve the decoding problem (find the most likely sequence of states), we evaluate the Viterbi algorithm

$$V_i(t) = \begin{cases} 0 & : \quad t = 0 \wedge i \neq S_I \\ 1 & : \quad t = 0 \wedge i = S_I \\ \max V_j(t-1) a_{ji} b_{ji}(y) & : \quad t > 0 \end{cases}$$

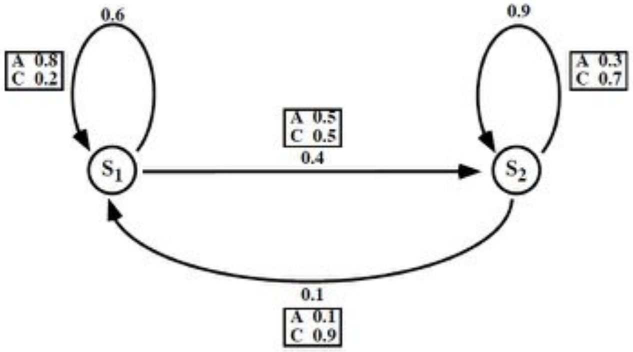
Where *$V_i(t)$* is the probability that the HMM is in state *i* after generating the sequence *$y_1, y_2, \ldots, y_t$*, following the *most probable path* in the HMM

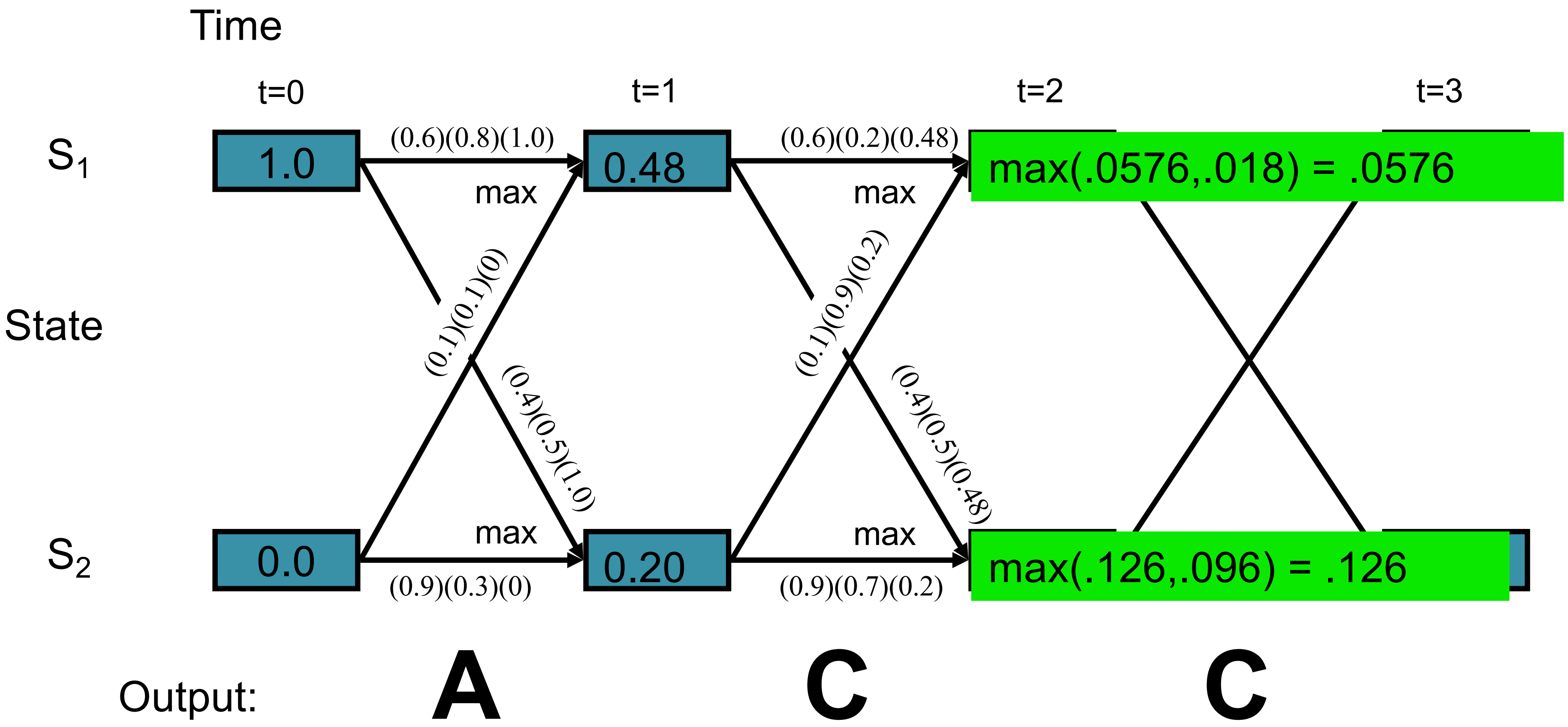
# A trellis for the Viterbi Algorithm



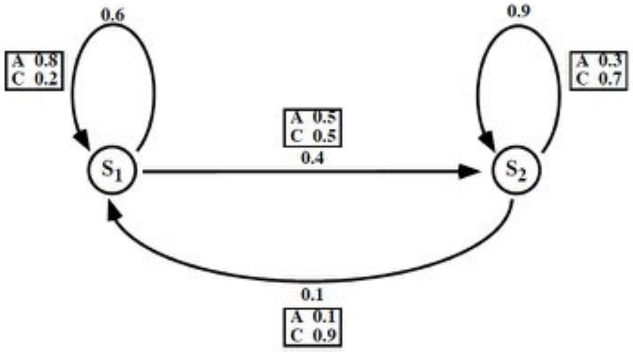


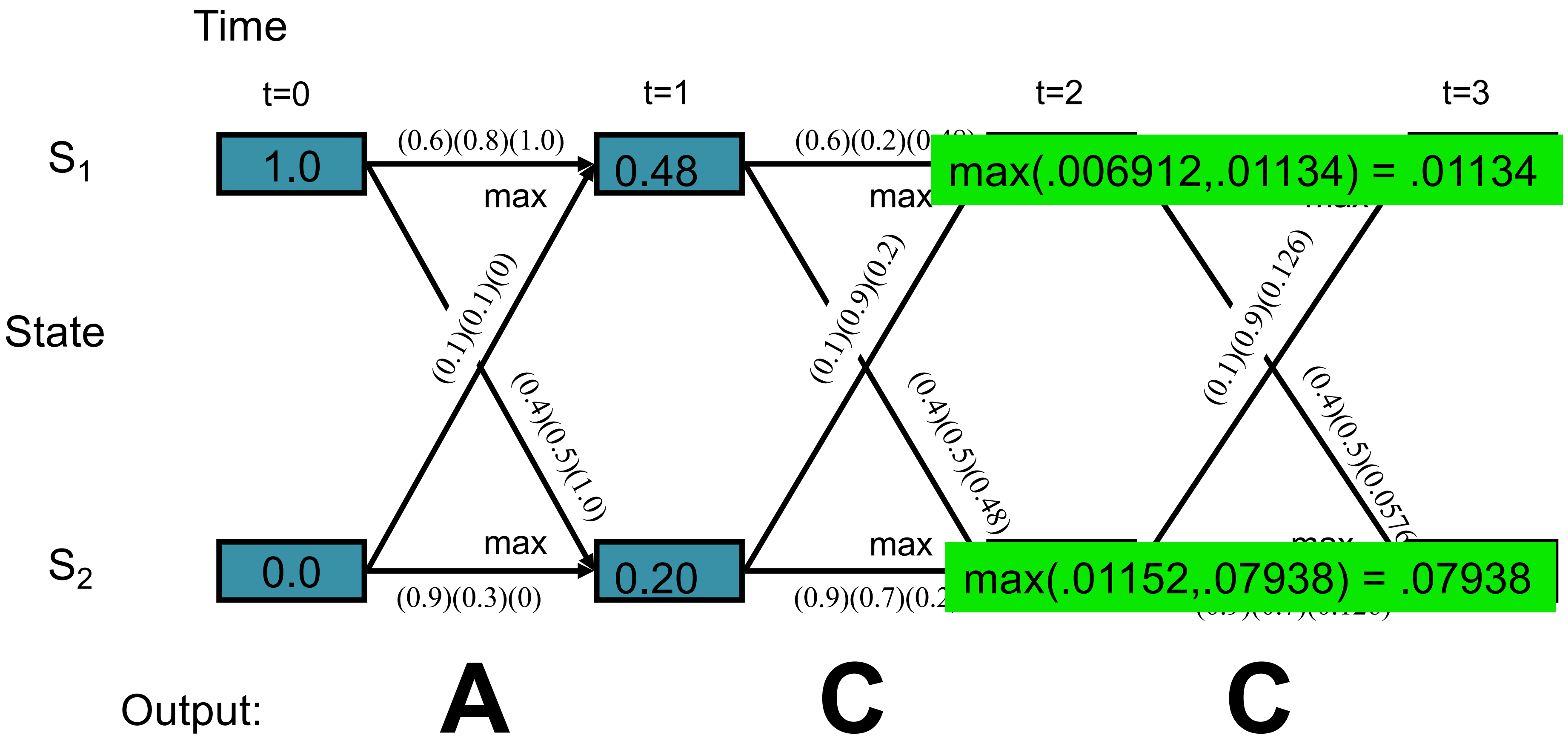
# A trellis for the Viterbi Algorithm





# A trellis for the Viterbi Algorithm



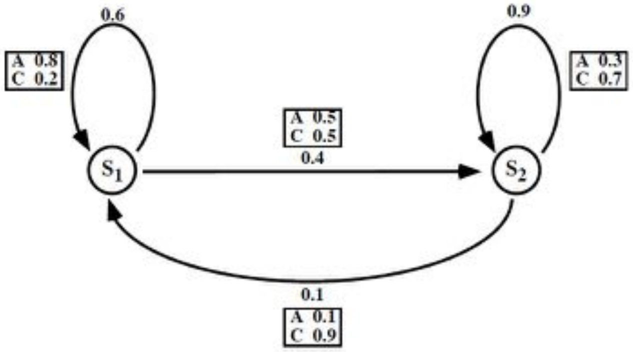


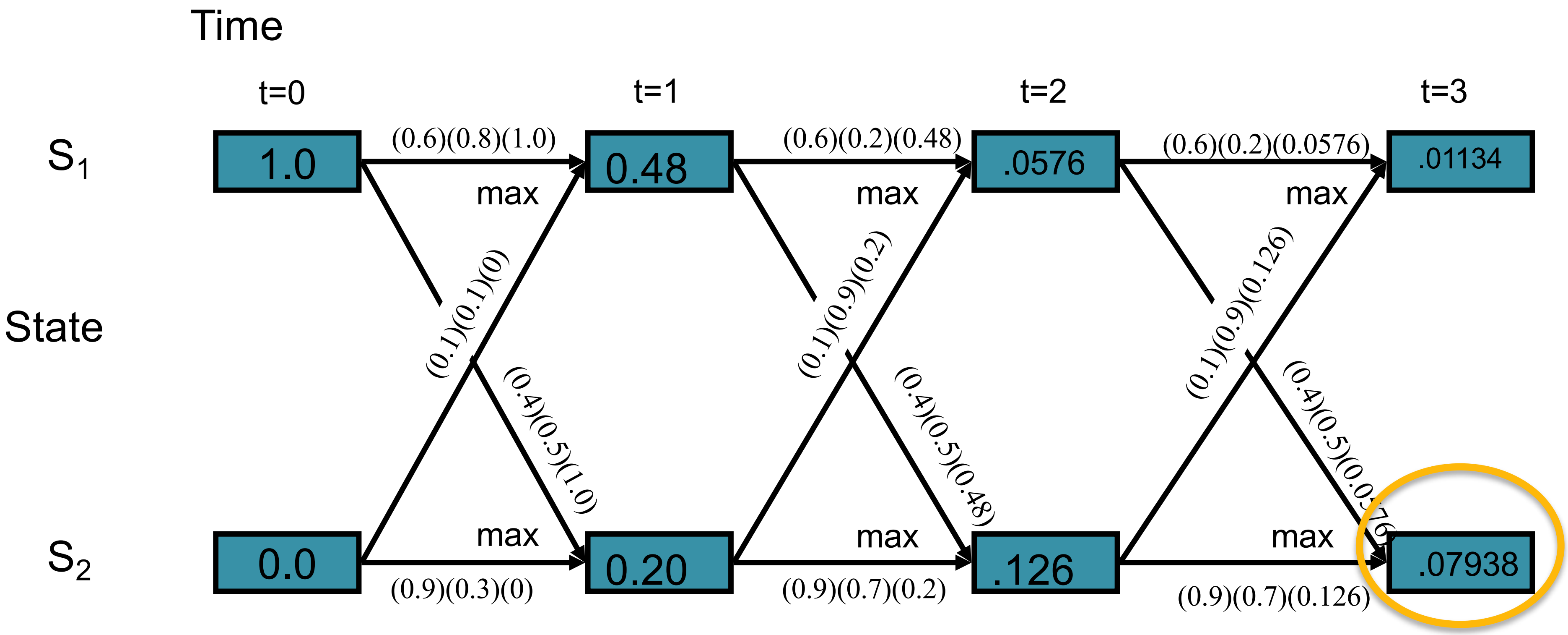
max(.006912,.01134) = .01134

max(.01152,.07938) = .07938

Output: **A** **C** **C**

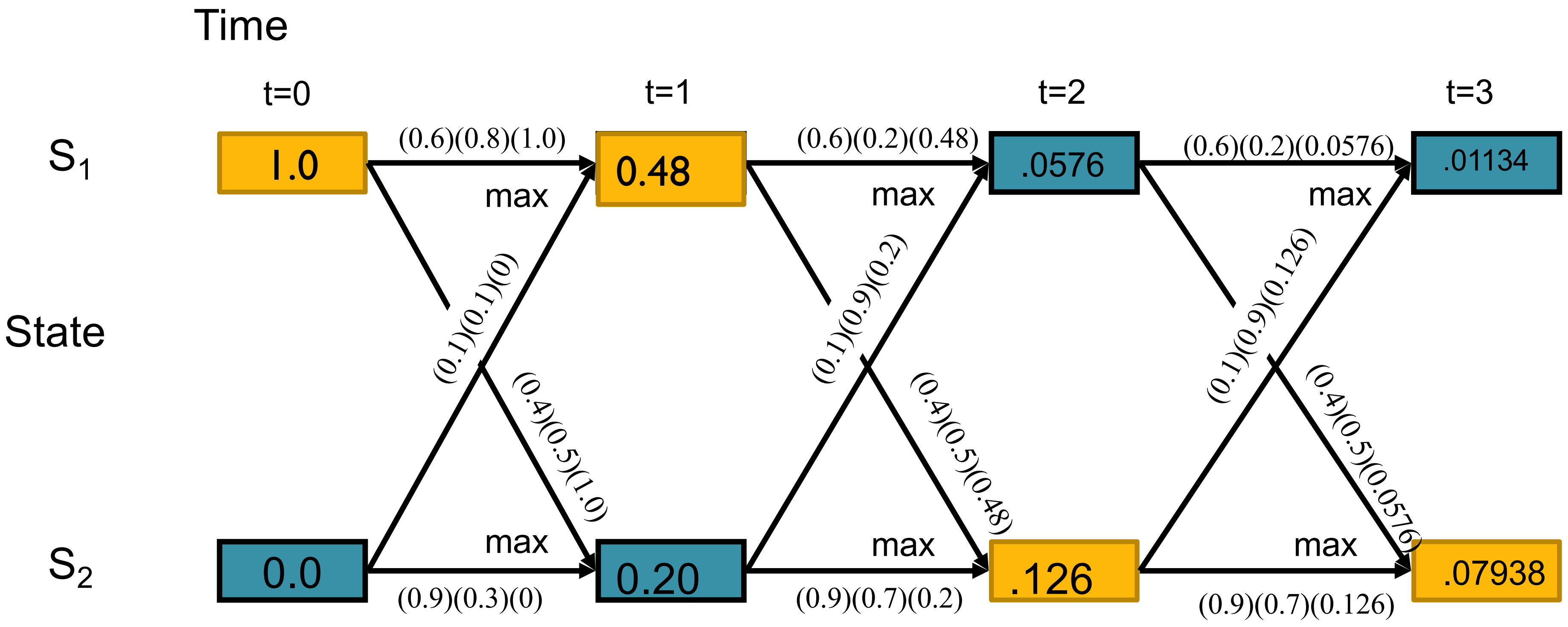
# A trellis for the Viterbi Algorithm





S2 is final state➔ the most probable sequence of states has a 7.9% probability

# A trellis for the Viterbi Algorithm



Time

t=0

t=1

t=2

t=3

$S_1$

State

$S_2$

1.0

0.48

.0576

.01134

0.0

0.20

.126

.07938

(0.6)(0.8)(1.0)

(0.6)(0.2)(0.48)

(0.6)(0.2)(0.0576)

max

max

max

(0.1)(0.1)(0)

(0.4)(0.5)(1.0)

(0.1)(0.9)(0.2)

(0.4)(0.5)(0.48)

(0.1)(0.9)(0.126)

(0.4)(0.5)(0.0576)

max

max

max

(0.9)(0.3)(0)

(0.9)(0.7)(0.2)

(0.9)(0.7)(0.126)

Output: **A** **C** **C**

Parse: $\mathbf{S_1}$ $\mathbf{S_2}$ $\mathbf{S_2}$

# Three classic HMM problems

3. **Learning**: given a model and a set of observed sequences, how do we set the model's parameters so that it has a high probability of generating those sequences?

- This is perhaps the most important, and most difficult problem.

- A solution to this problem allows us to determine all the probabilities in an HMMs by using an ensemble of training data

# Learning in HMMs:

- The learning algorithm uses Expectation-Maximization (E-M)
  - Also called the Forward-Backward algorithm
  - Also called the Baum-Welch algorithm

- In order to learn the parameters in an "empty" HMM, we need:
  - The topology of the HMM
  - Data - the more the better
  - Start with a random (or naïve) probability, repeat until converges

# Gene Finding Overview

- Prokaryotic gene finding distinguishes real genes and random ORFs
  - Prokaryotic genes have simple structure and are largely homogenous, making it relatively easy to recognize their sequence composition
- Eukaryotic gene finding identifies the genome-wide most probable gene models (set of exons)
  - "Probabilistic Graphical Model" to enforce overall gene structure, separate models to score splicing/transcription signals
  - Accuracy depends to a large extent on the quality of the training data