Functional Genomics 2: Gene Annotation

Michael Schatz

Oct 16, 2019

Lecture 14: Computational Biomedical Research

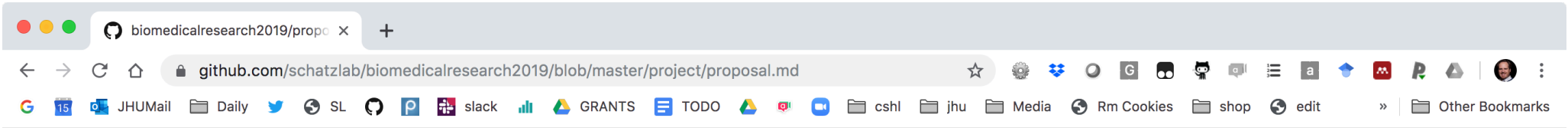
Project Pitches

Student	Topic
Mary Joseph	Ethnic origins
Christian Seremetis	Ethnic origins
Gautam Prabhu	Disease risk: pathways, epistatic mutations
Joanna Guo	Disease risk: pathways, graph analysis
David Yang	Disease risk: classifiers
Kavya Tumkur	Disease risk: classifiers
Richard Xu	Disease risk: SVs, classifiers

Project Timeline

Week	Date	Deliverable
1	Oct 14	Decide teams
2	Oct 21	Abstract + Presentation
3	Oct 28	
4	Nov 4	Intern Report
5	Nov 11	
6	Nov 18	Peer code review
7	Nov 25	<Thanksgiving>
8	Dec 2	In class presentation
9	Dec 9	
10	Dec 16	Final Report Due

Project Proposal



Project Proposal

Assignment Date: Wednesday Oct 16, 2019
Due Date: Tuesday, Oct 22, 2019 @ 11:59pm

Project Teams:

Team A:

Mary Joseph, Christian Seremetis, Gautam Prabhu, Joanna Guo

Team B:

David Yang, Kavya Tumkur, Richard Xu

Proposal

The proposal should have the following components:

- Name of your team
- List of team members and email addresses
- Short title for your proposal
- 1 - 2 paragraph description of what you hope to do and how you will do it
- References to 2 to 3 relevant papers
- References/URLs to datasets that you will be studying (Note you can also use simulated data)
- Please add a note if you need me to sponsor you for a MARCC account (high RAM, GPUs, many cores, etc)

Submit the proposal as a single page PDF on GradeScope (each team member should submit the same PDF). Then in class on Wednesday Oct 23, each team will orally present their plan.

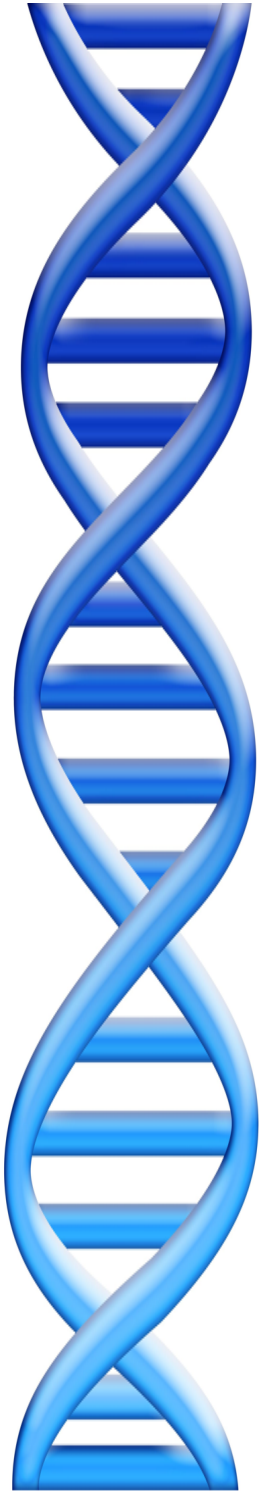
Recommended outline for team presentation:

Goal: Genome Annotations

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

Goal: Genome Annotations

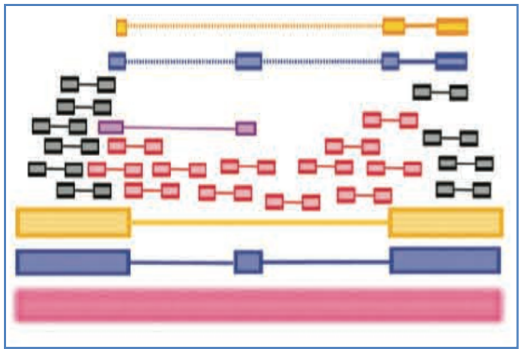
Gene!



Outline

1. Experimental: RNAseq
2. Homology: Alignment to other genomes
3. Prediction: "Gene Finding"

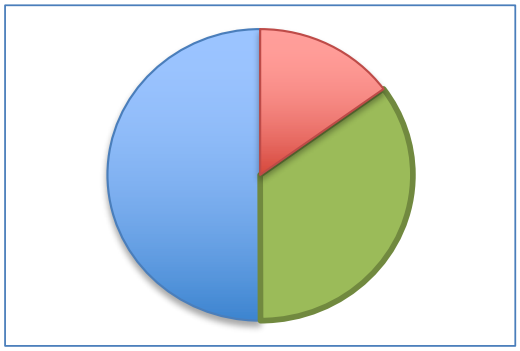
RNA-seq Challenges



Challenge 1: Eukaryotic genes are spliced

Solution: Use a spliced aligner, and assemble isoforms

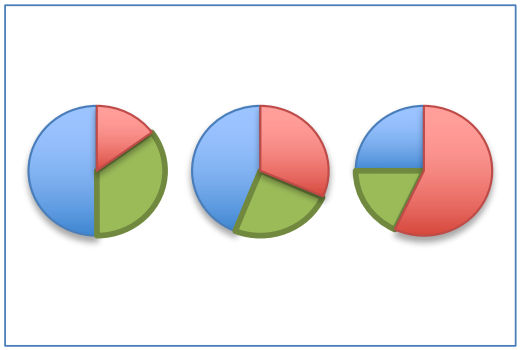
TopHat: discovering spliced junctions with RNA-Seq.
Trapnell et al (2009) *Bioinformatics*. 25:0 1105-1111



Challenge 2: Read Count != Transcript abundance

Solution: Infer underlying abundances (e.g. TPM)

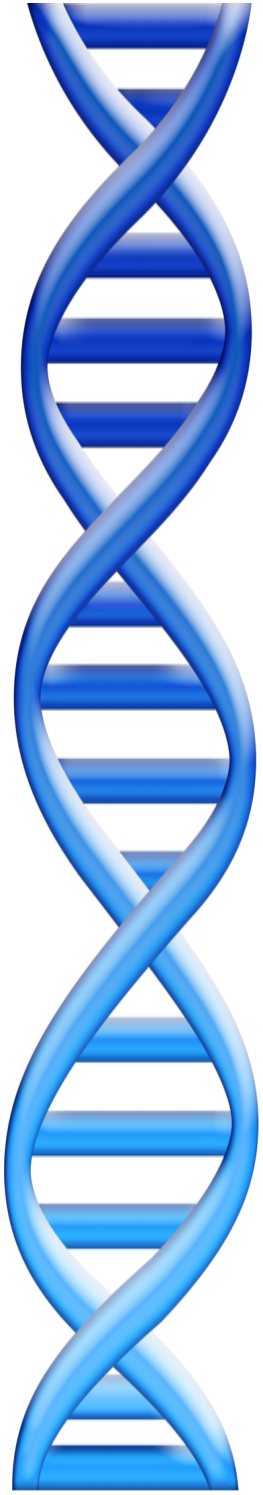
Transcript assembly and quantification by RNA-seq
Trapnell et al (2010) *Nat. Biotech.* 25(5): 511-515



Challenge 3: Transcript abundances are stochastic

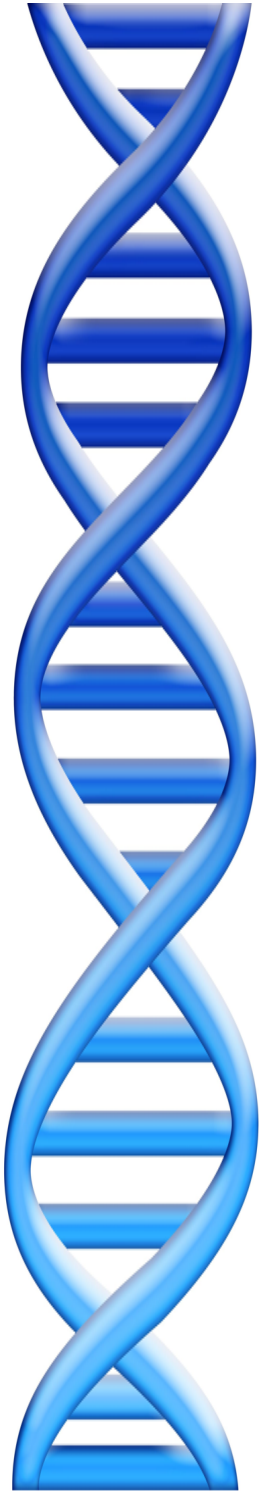
Solution: Replicates, replicates, and more replicates

RNA-seq differential expression studies: more sequence or more replication?
Liu et al (2013) *Bioinformatics.* doi:10.1093/bioinformatics/btt688



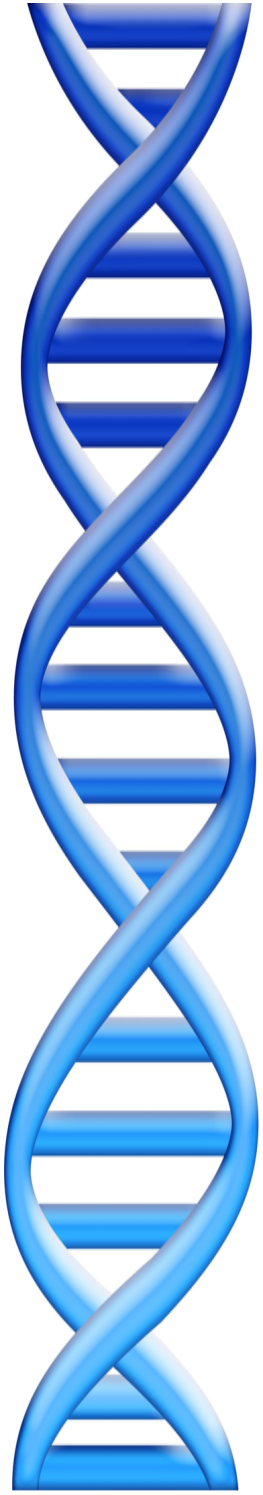
Outline

1. Experimental: RNAseq
   - ☺ Direct evidence for expression!
     - Including novel genes within a species
   - ☹ Typical tissues only express 25% to 50% of genes
     - Many genes are restricted to very particular cell types, developmental stages, or stress conditions
     - Our knowledge of alternative splicing is very incomplete
   - ☹ Can resolve gene structure, but nothing about gene function
     - Co-expression is sometimes a clue, but often incomplete



Outline

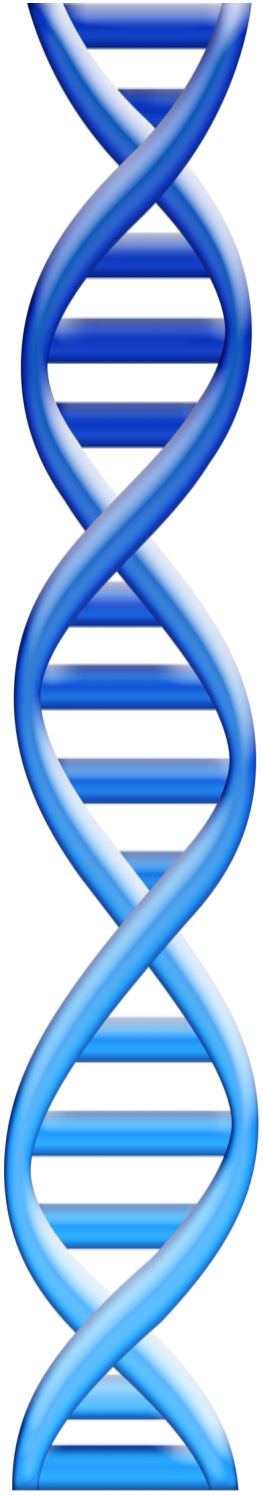
1. Experimental: RNAseq
2. Homology: Alignment to other genomes
3. Prediction: "Gene Finding"



Outline

2. Homology: Alignment to other genomes

- :-/ Indirect evidence for expression
  - Works well for familiar species, but more limited for unexplored clades
  - Relatively few false positives, but many false negatives
- ☺ Universal across tissues (and species)
  - Proteins often have highly conserved domains, whereas genome/transcript may have many mutations (especially "wobble" base)
- :-/ Transfer gene function across species
  - Reciprocal best blast hit a widely used heuristic
  - Often works, but examples where single base change leads to opposite function



Outline

1. Experimental: RNAseq
2. Homology: Alignment to other genomes
3. Prediction: "Gene Finding"

Bacterial Gene Finding and Glimmer

(also Archaeal and viral gene finding)

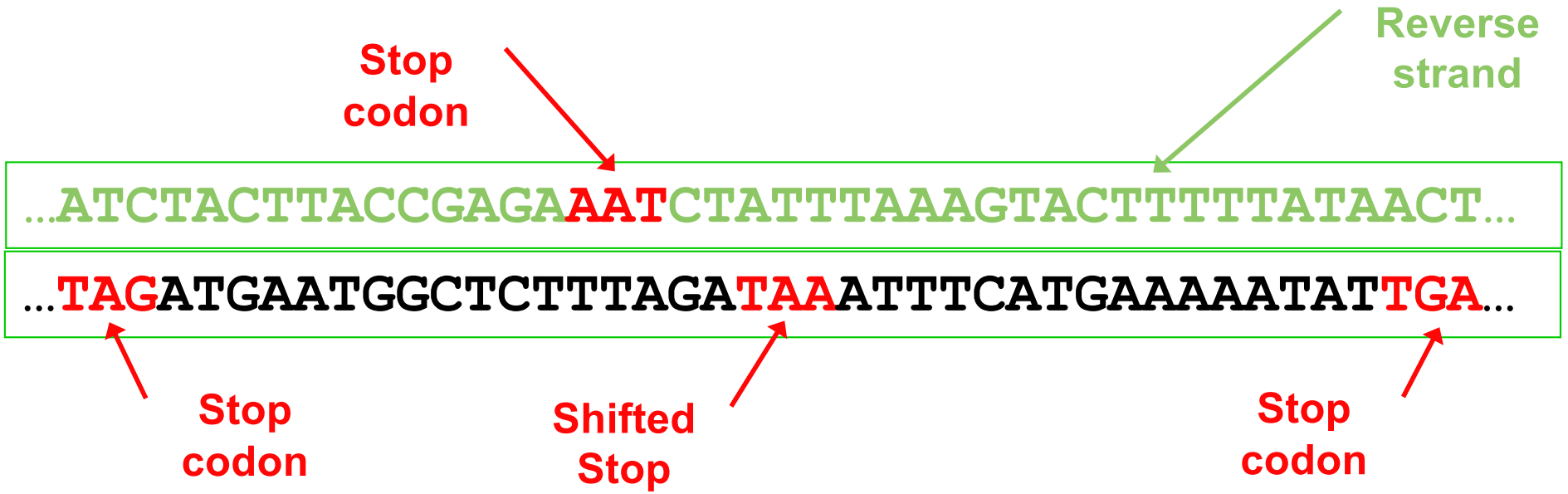
Arthur L. Delcher and Steven Salzberg
Center for Bioinformatics and Computational Biology
Johns Hopkins University School of Medicine

Step One

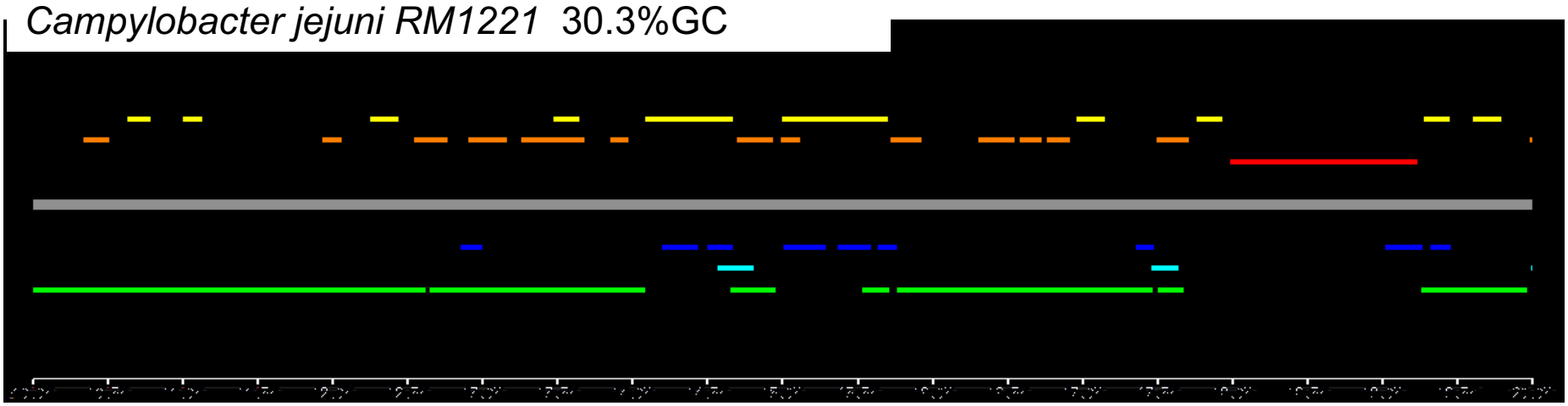
- Find open reading frames (ORFs).

Step One

- Find open reading frames (ORFs).



- But ORFs generally overlap …

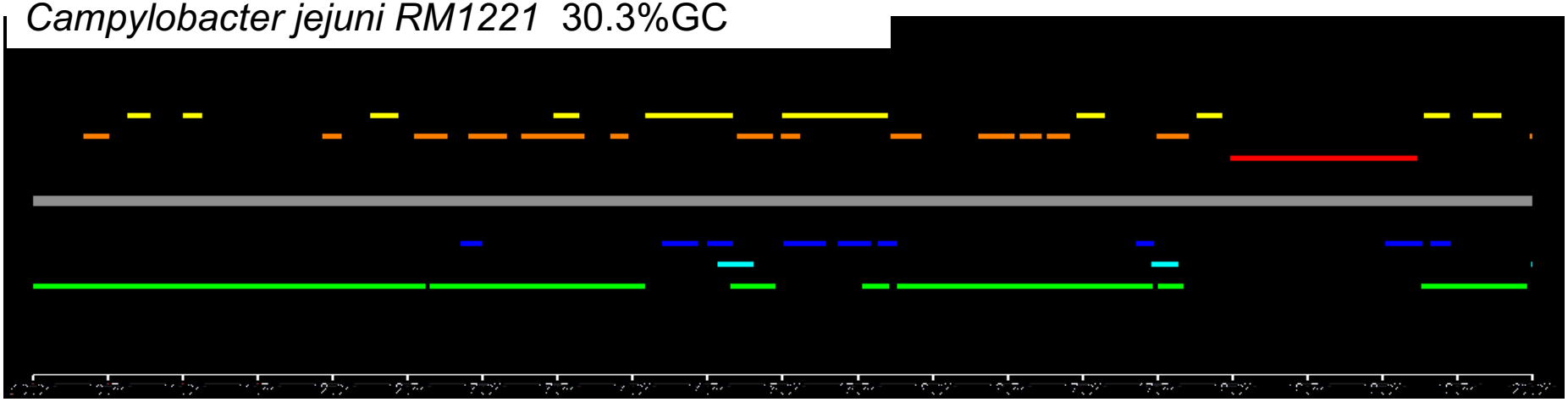


All ORFs longer than 100bp on both strands shown
- color indicates reading frame

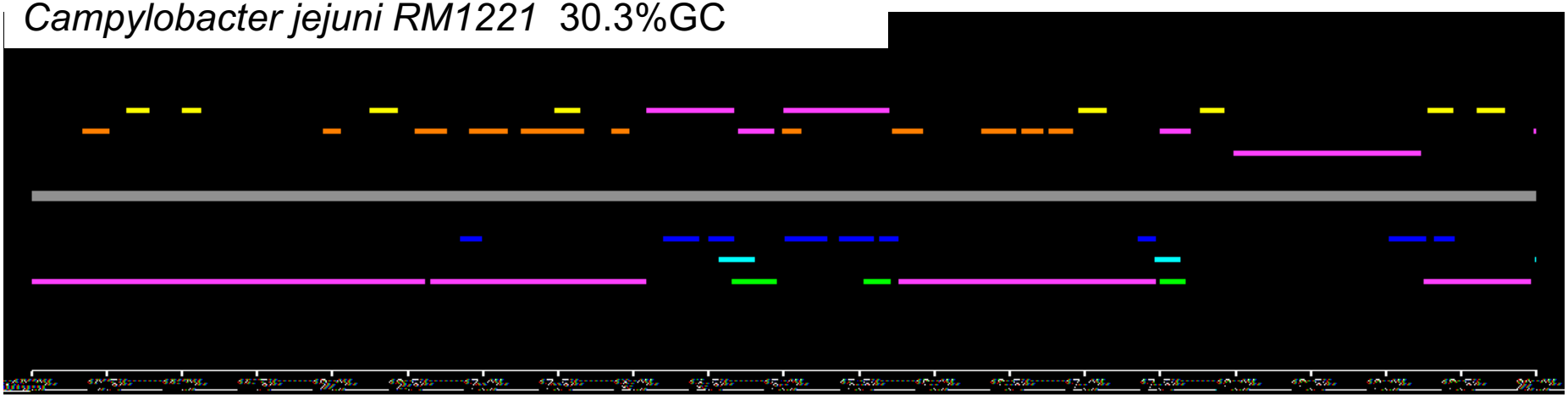
Note the low GC content
- many A+T -> many stop codons (TAA/TAG/TGA)

All genes are ORFs but not all ORFs are genes
- Longest ORFs likely to be protein-coding genes

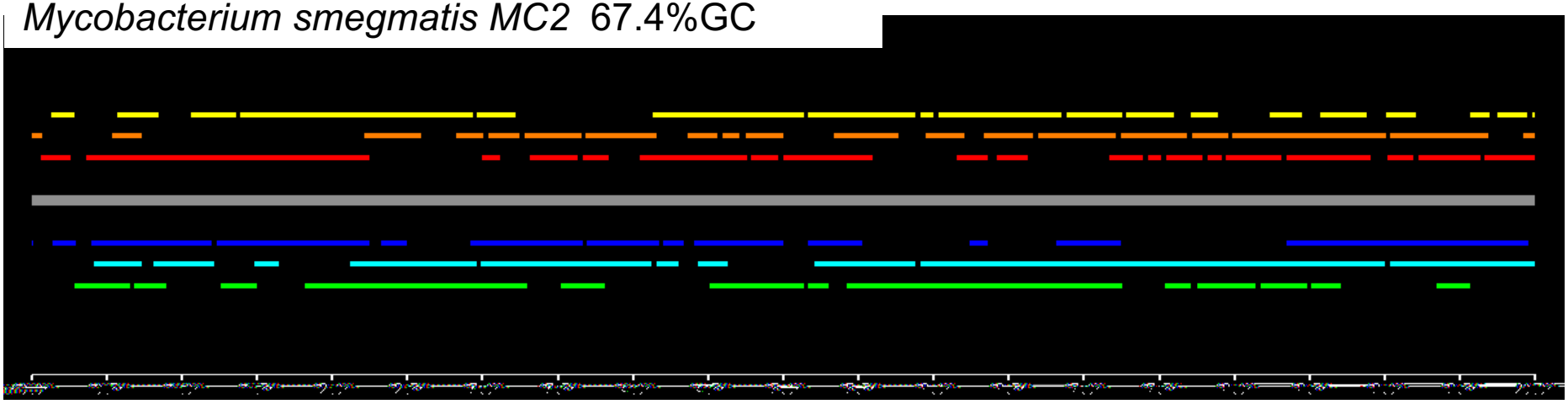
Campylobacter jejuni RM1221 30.3%GC



Campylobacter jejuni RM1221 30.3%GC

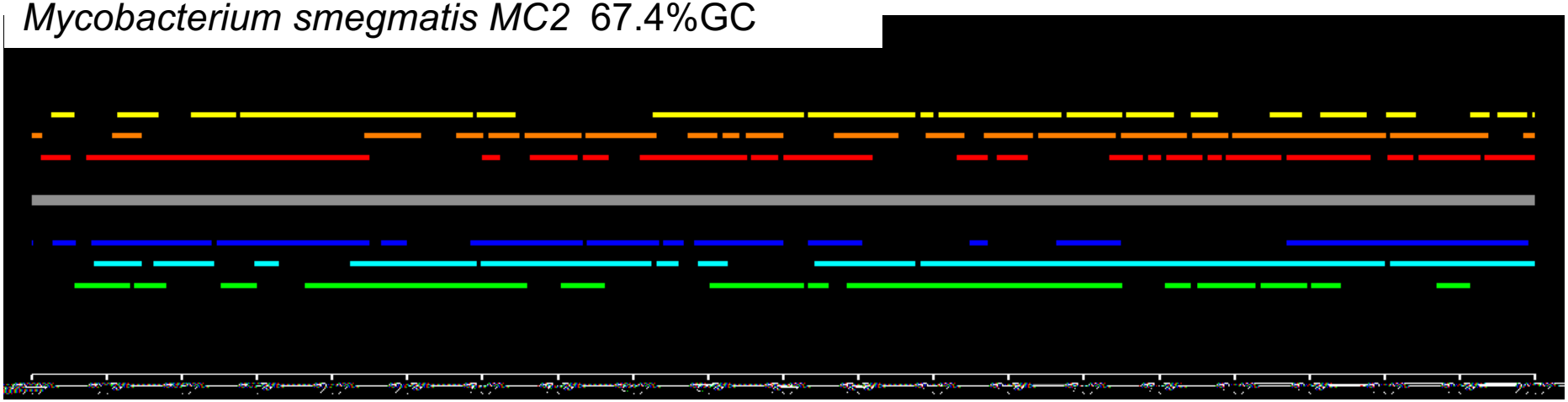


Mycobacterium smegmatis MC2 67.4%GC

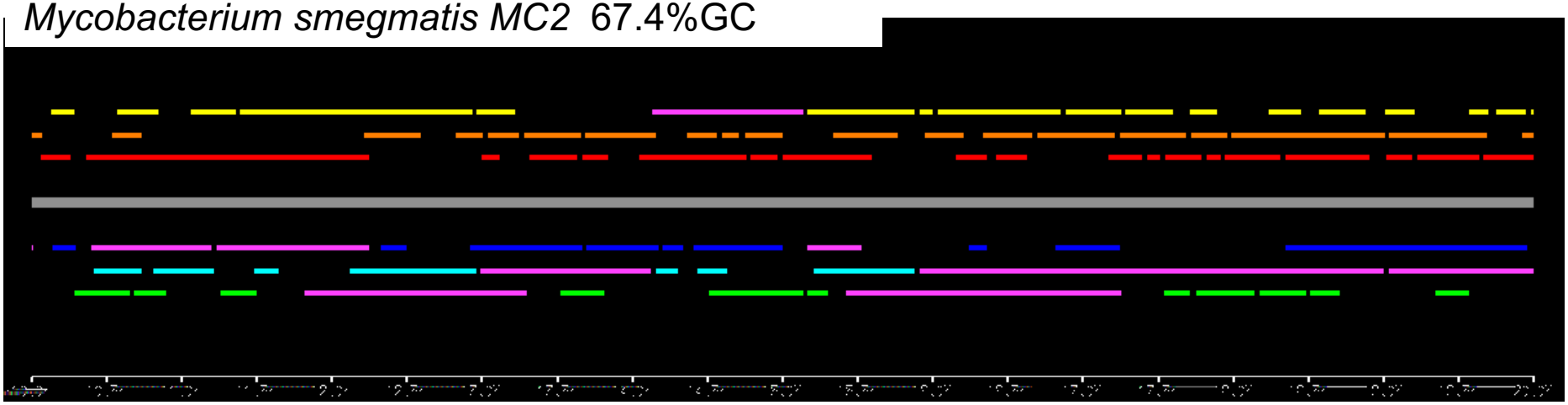


Note what happens in a high-GC genome

Mycobacterium smegmatis MC2 67.4%GC



Mycobacterium smegmatis MC2 67.4%GC



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

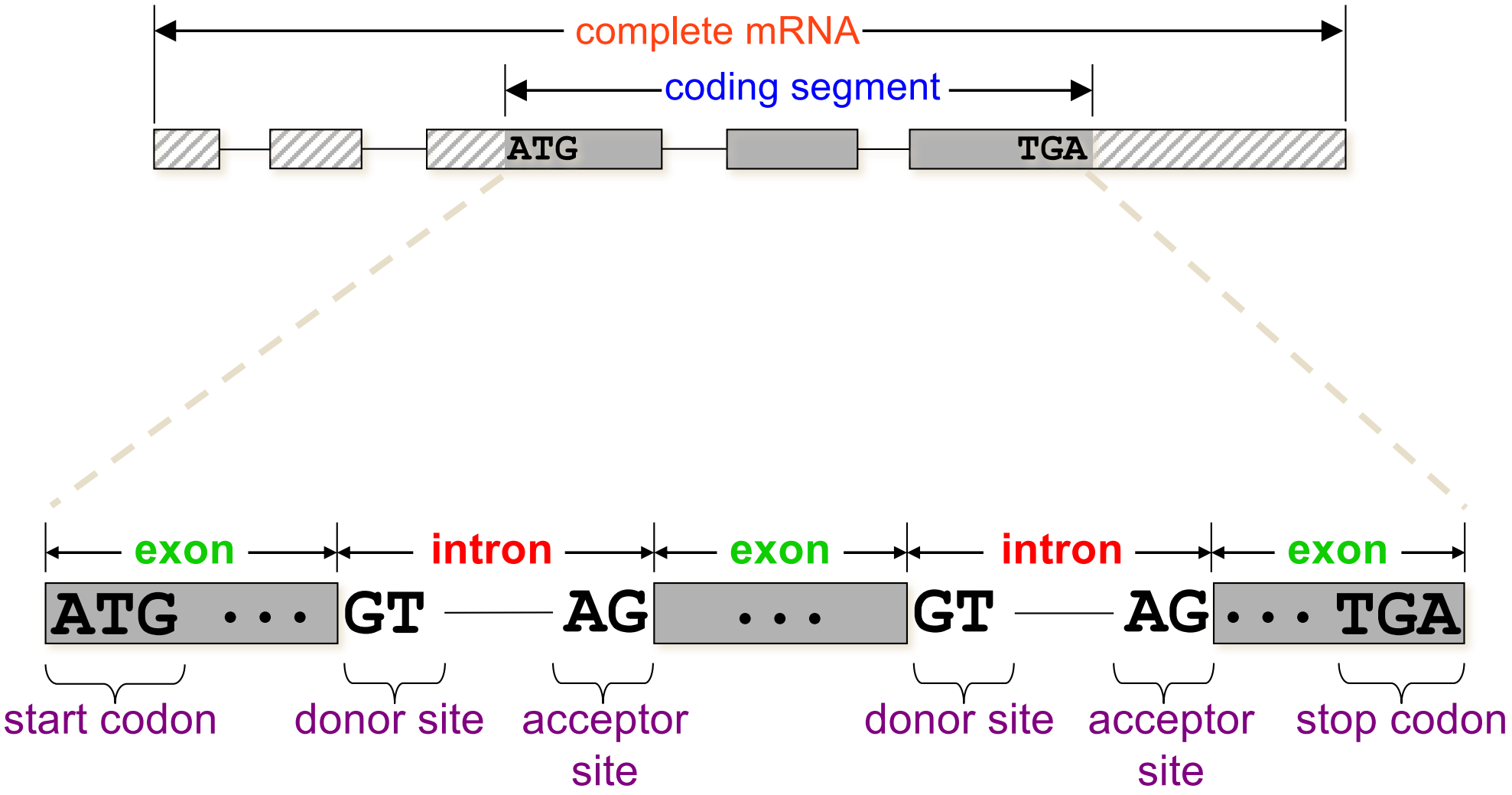
Overview of Eukaryotic Gene Prediction

CBB 231 / COMPSCI 261

W.H. Majoros

Duke UNIVERSITY

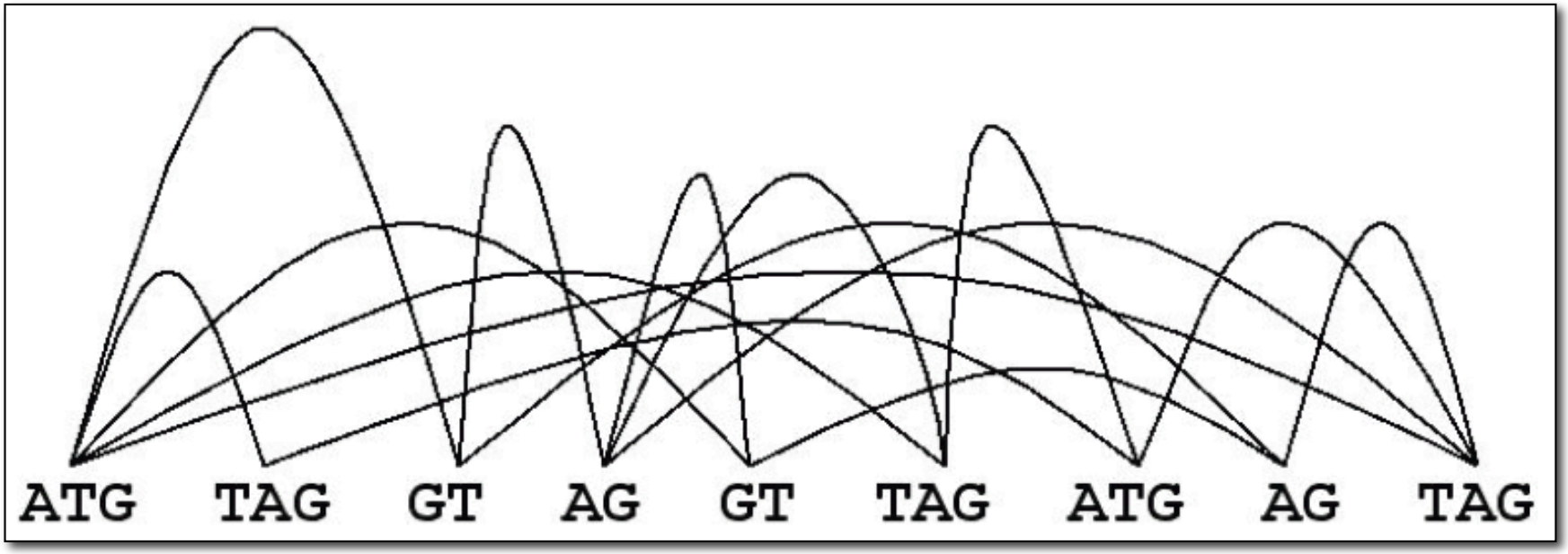
Eukaryotic Gene Syntax



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*:

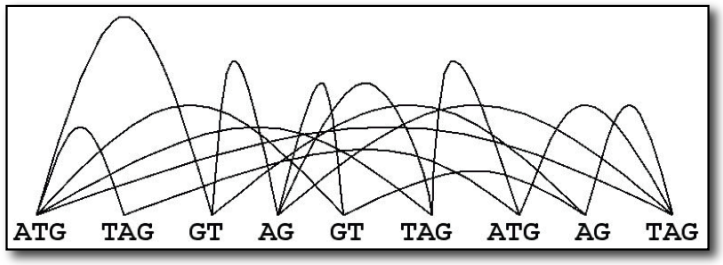


An ORF graph represents all possible *gene parses* (and their scores) for a given set of putative signals. A *path* through the graph represents a single gene parse.

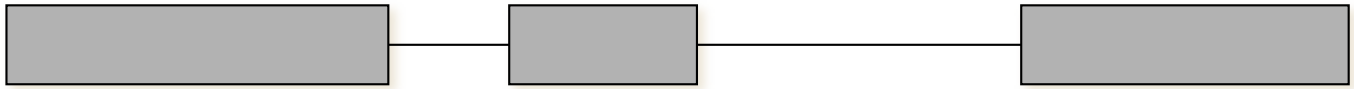
Conceptual Gene-finding Framework

```
TATTCCGATCGATCGATCTCTCTAGCGTCTACG
CTATCATCGCTCTCTATTATCGCGCGATCGTCG
ATCGCGCGAGAGTATGCTACGTCGATCGAATTG
```

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

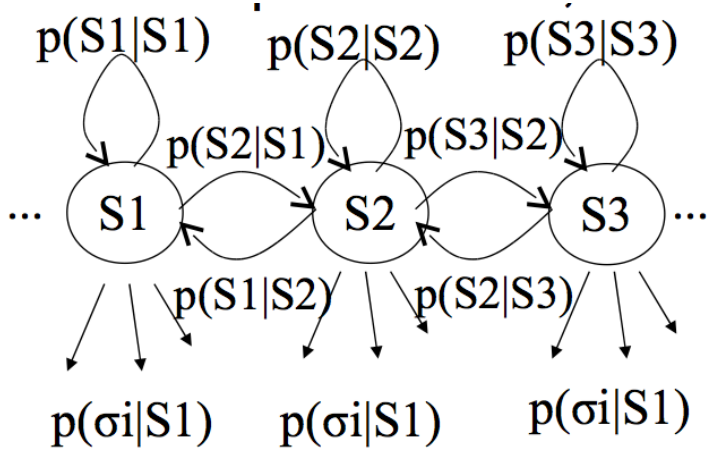


find highest-scoring path through ORF graph; interpret path as a gene parse = gene structure



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



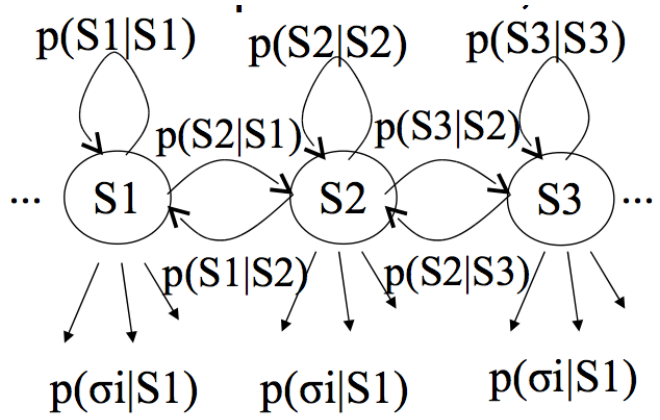
  - A set of emission characters
    - E={H,T} for coin tossing
    - E={1,2,3,4,5,6} for dice tossing
    - E={A,C,G,T} for DNA

  - 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
    - $P(s_i=\text{Fair}|s_{i-1}=\text{Fair}) = .9$, $P(s_i=\text{Bias}|s_{i-1} = \text{Fair}) .1$
    - $P(s_i=\text{Exon} | s_{i-1}=\text{Intergenic})$, …

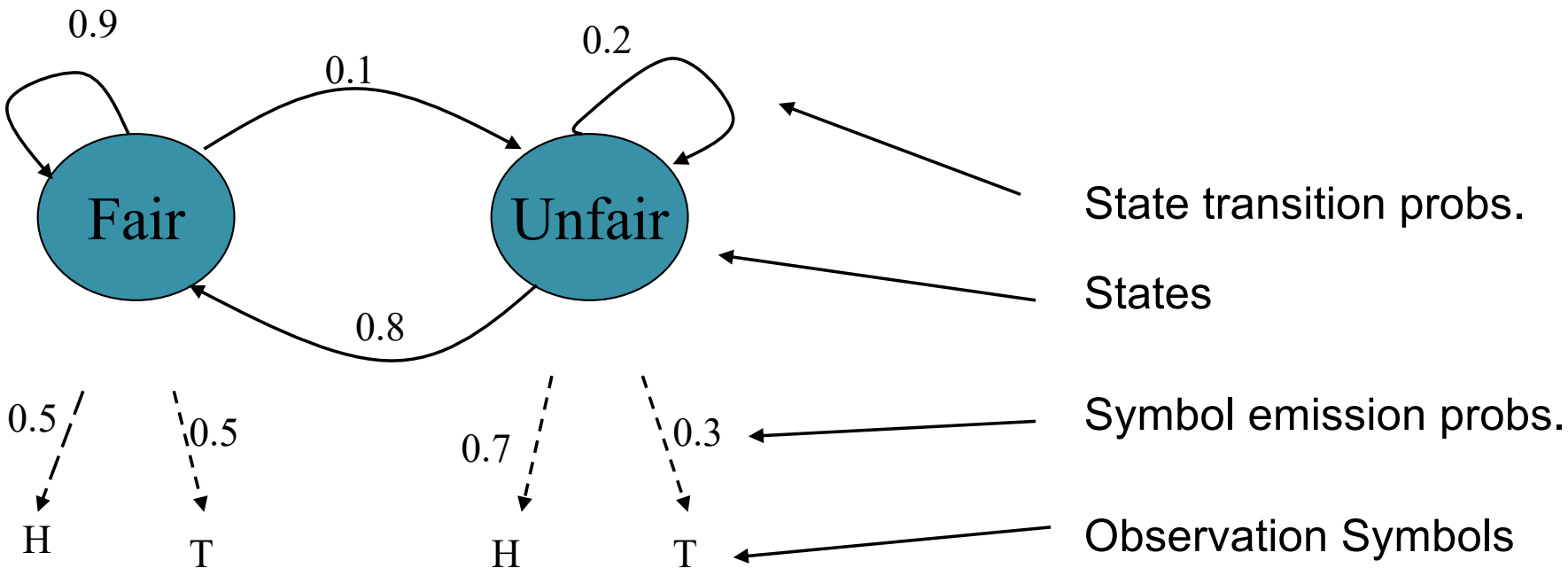
Why Hidden?

- 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

HMM Example - Casino Coin



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

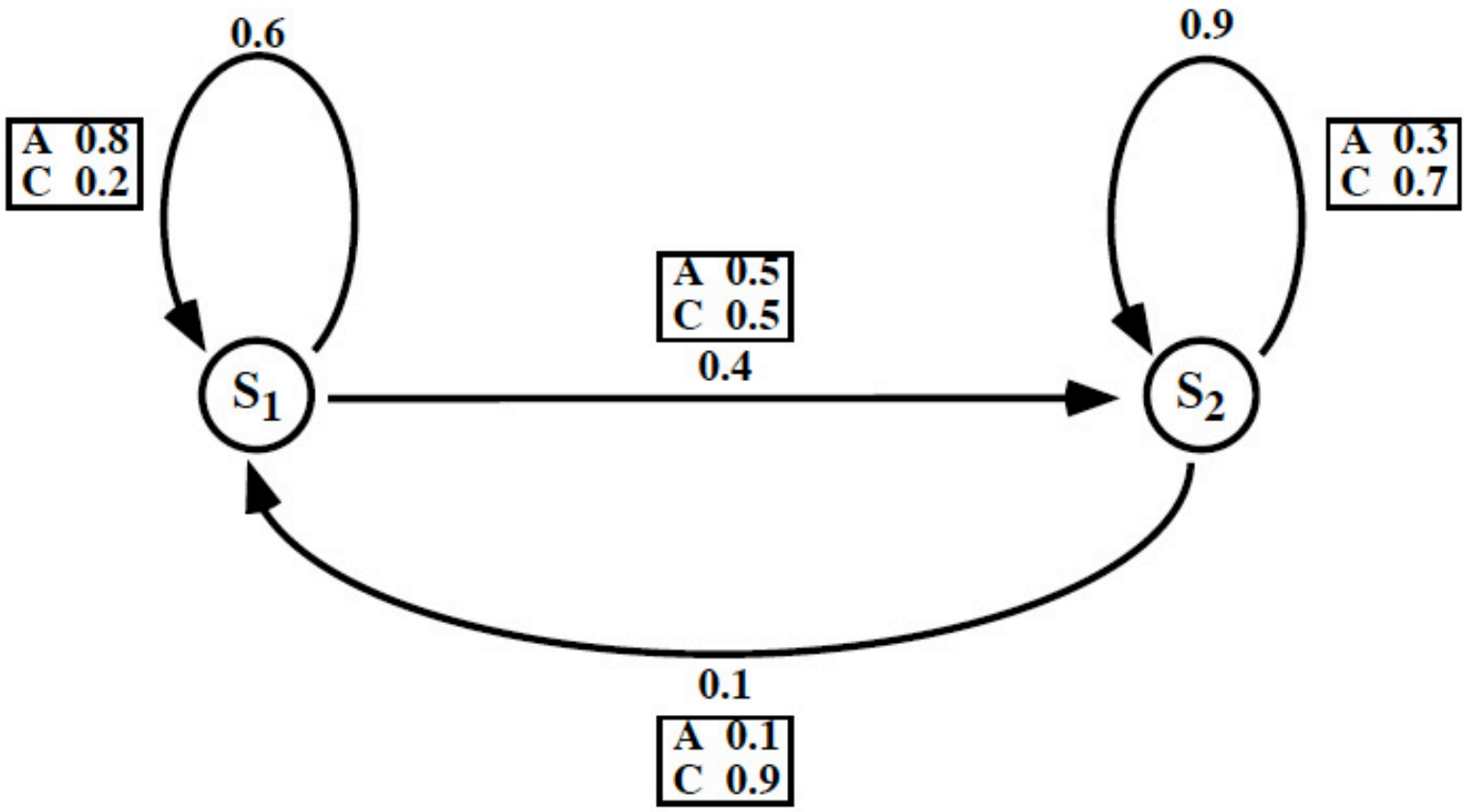
- Example: we might have a set of HMMs representing protein families -> pick the model with the best score

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*

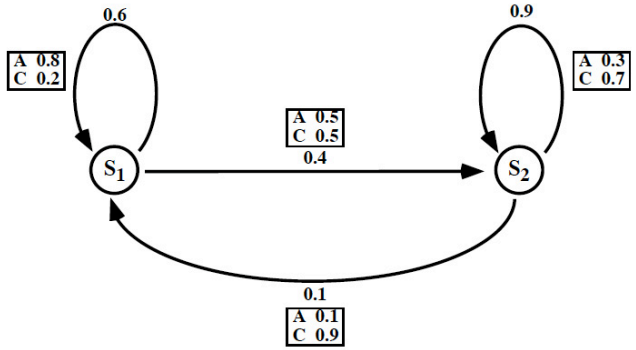
- Filling in the trellis will give tell us the probability that the HMM generated the data by finding all possible paths that could do it
  - Especially useful to evaluate from which models, a given sequence is most likely to have originated

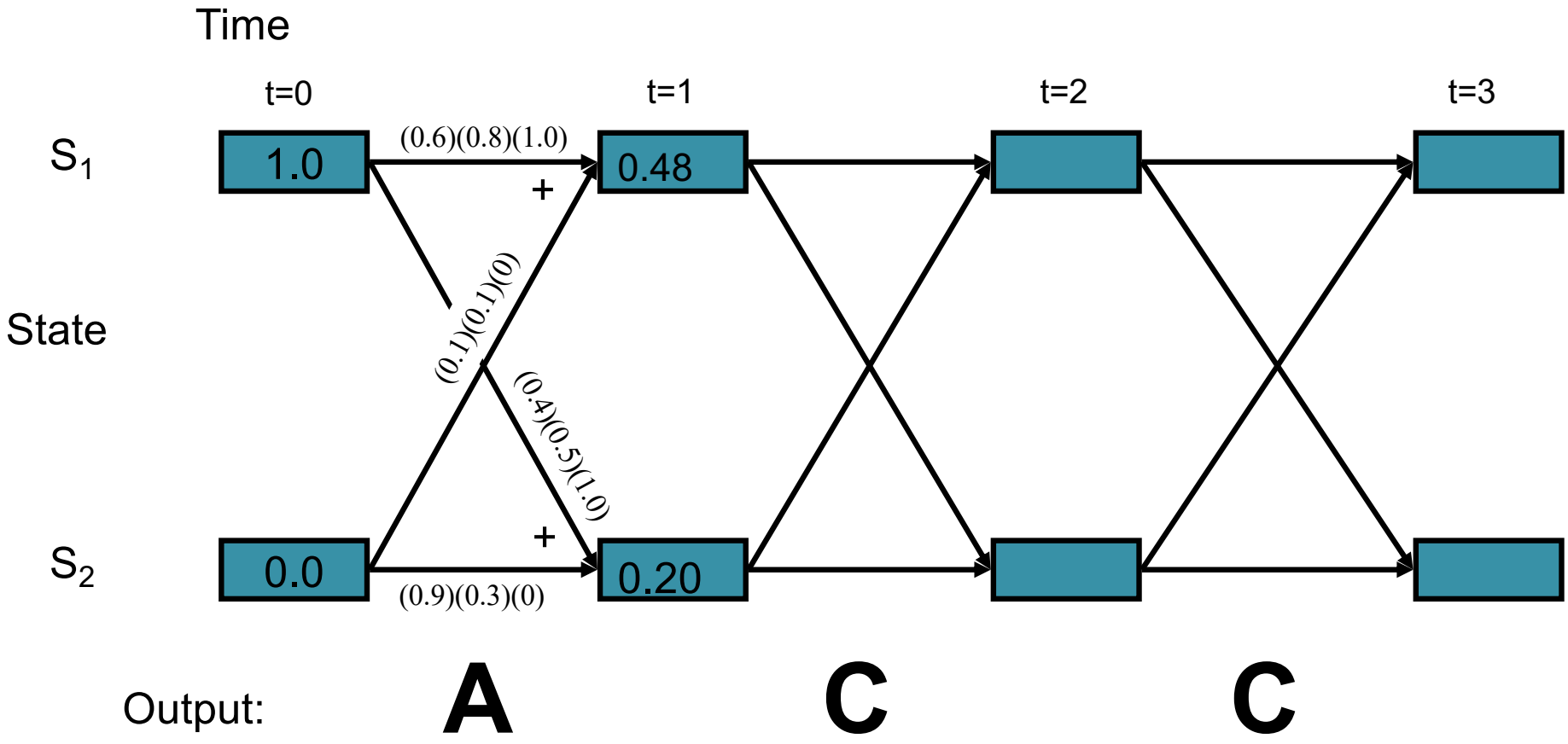
Our sample HMM



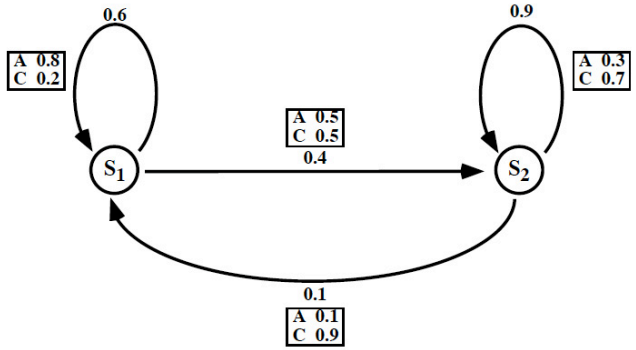
Let S_1 be initial state, S_2 be final state

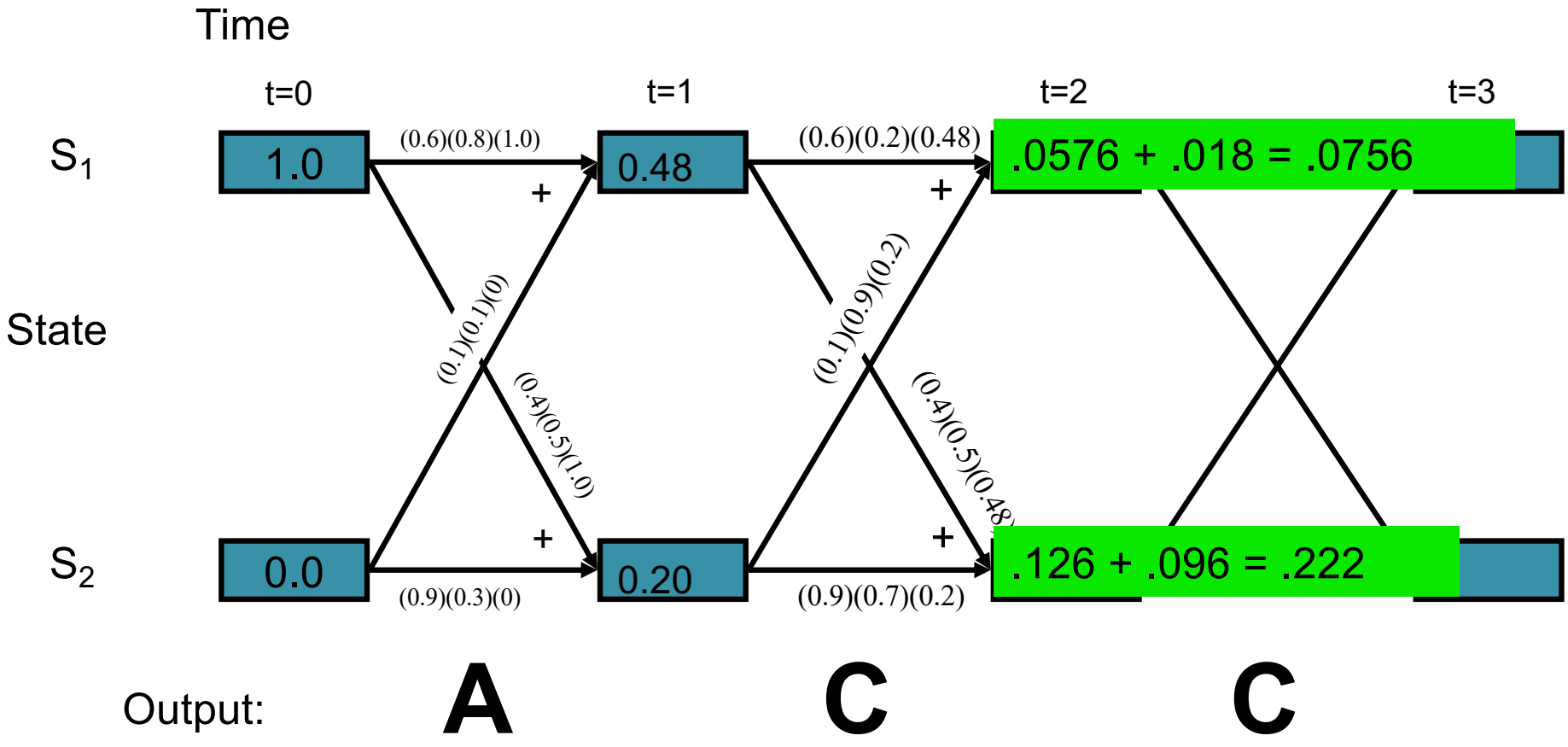
A trellis for the Forward Algorithm



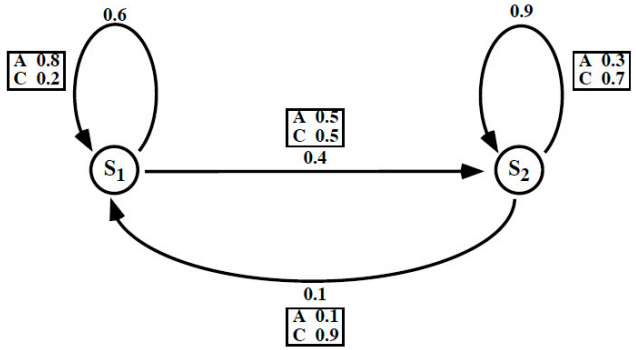


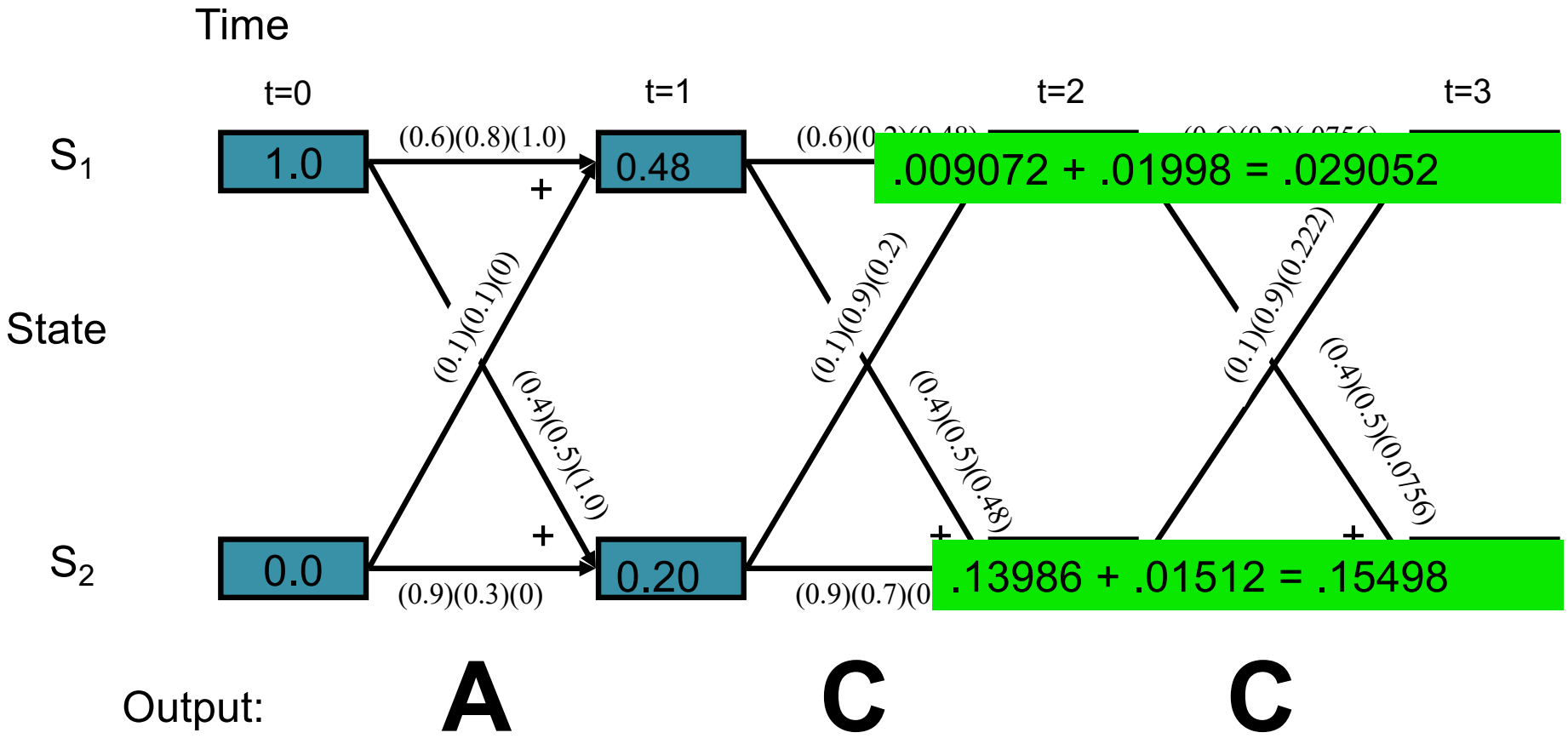
A trellis for the Forward Algorithm



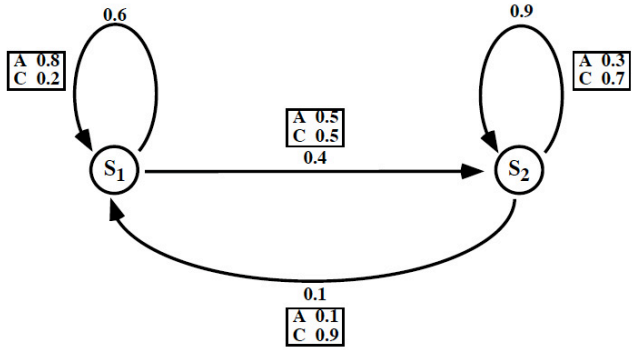


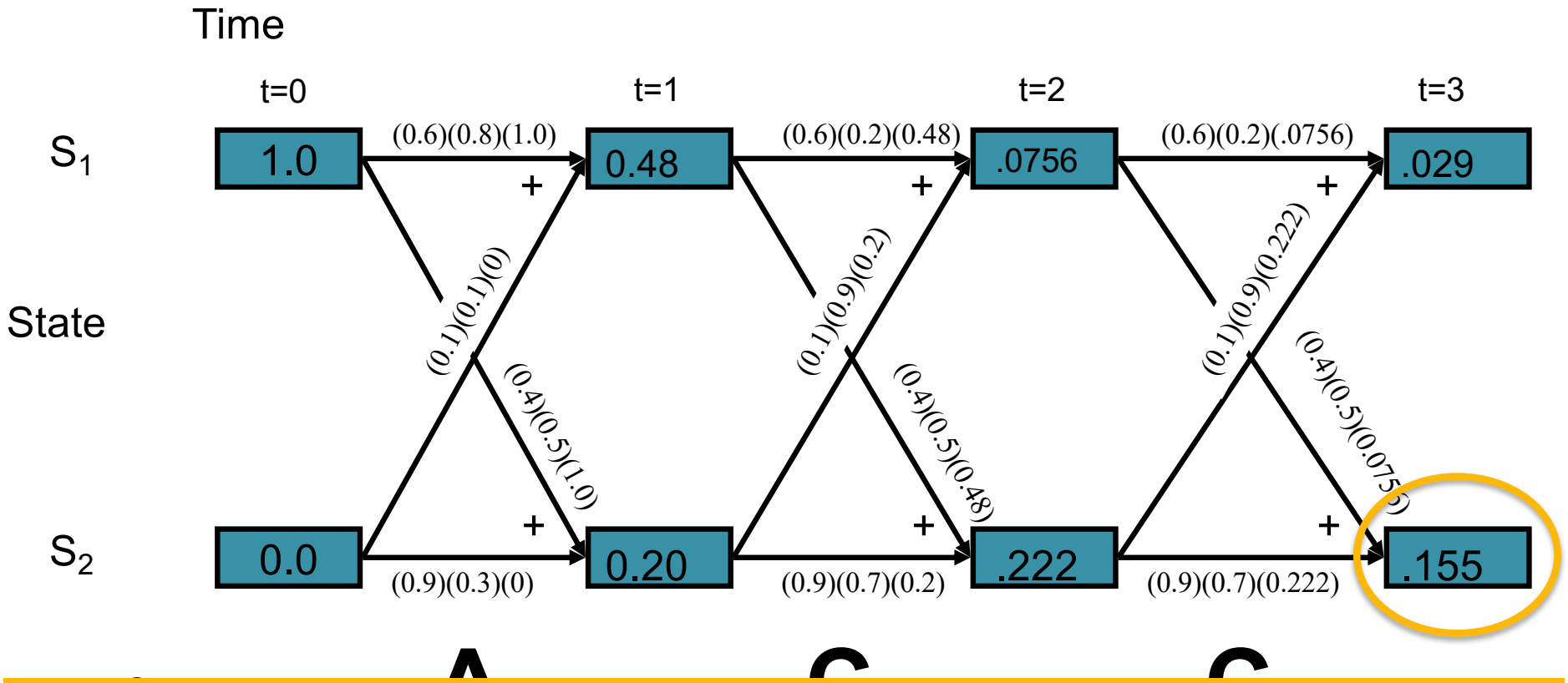
A trellis for the Forward Algorithm





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

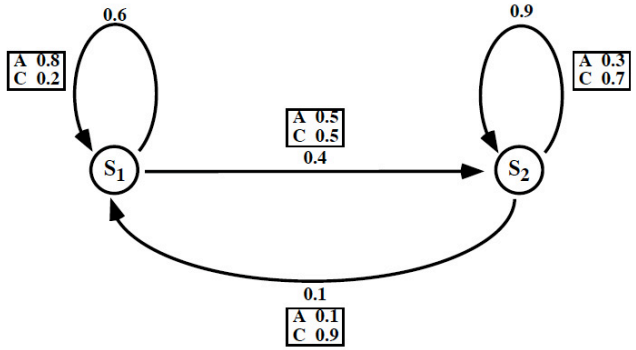
Solving the Decoding Problem: The Viterbi algorithm

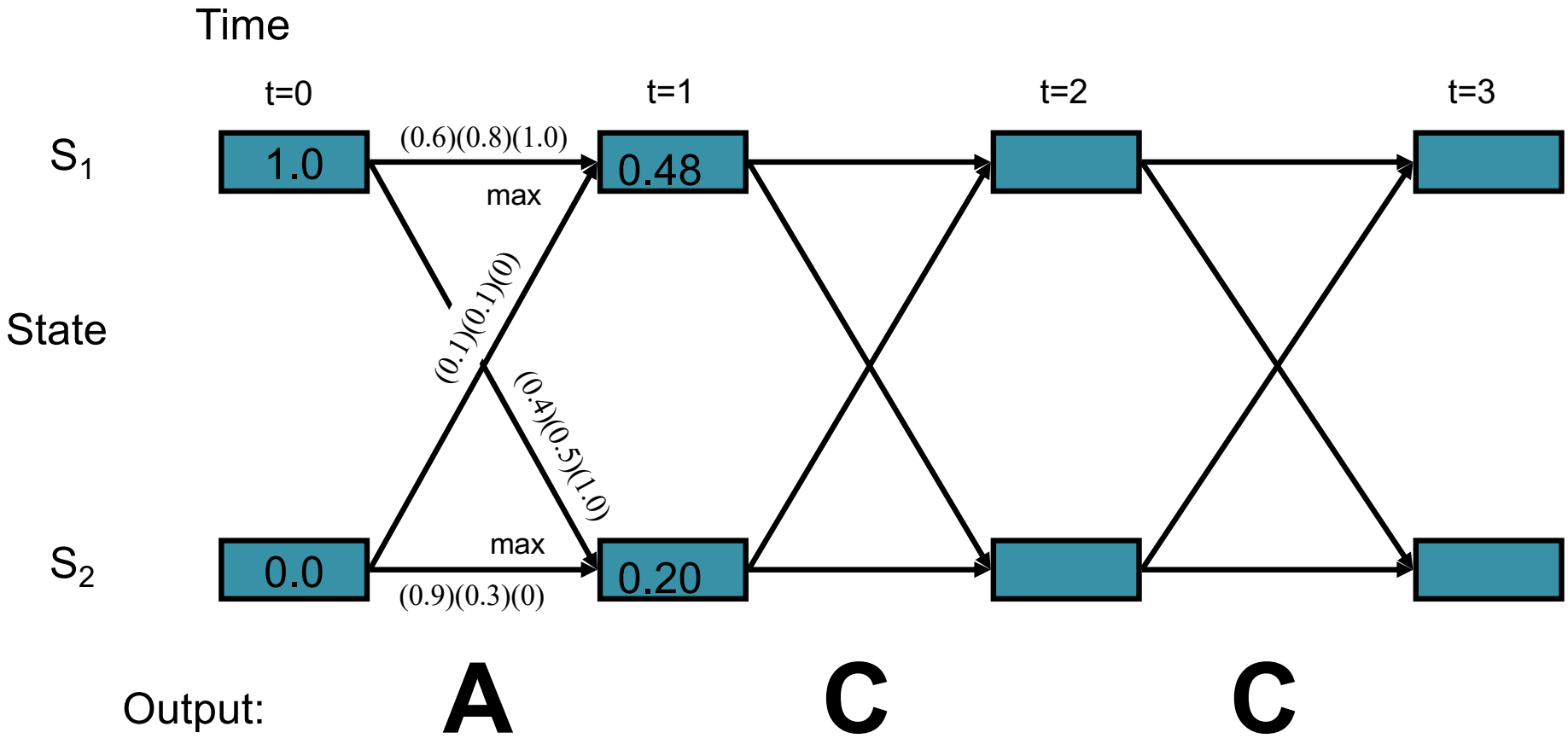
- To solve the decoding problem (find the most likely sequence of states), we evaluate the Viterbi algorithm

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

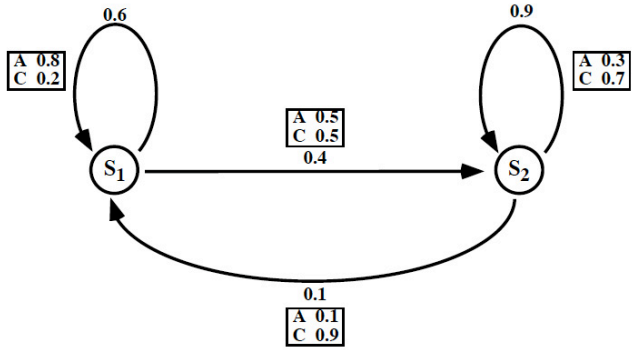
Where $V_i(t)$ is the probability that the HMM is in state i after generating the sequence $y_1, y_2, \ldots, y_t$, following the *most probable path* in the HMM

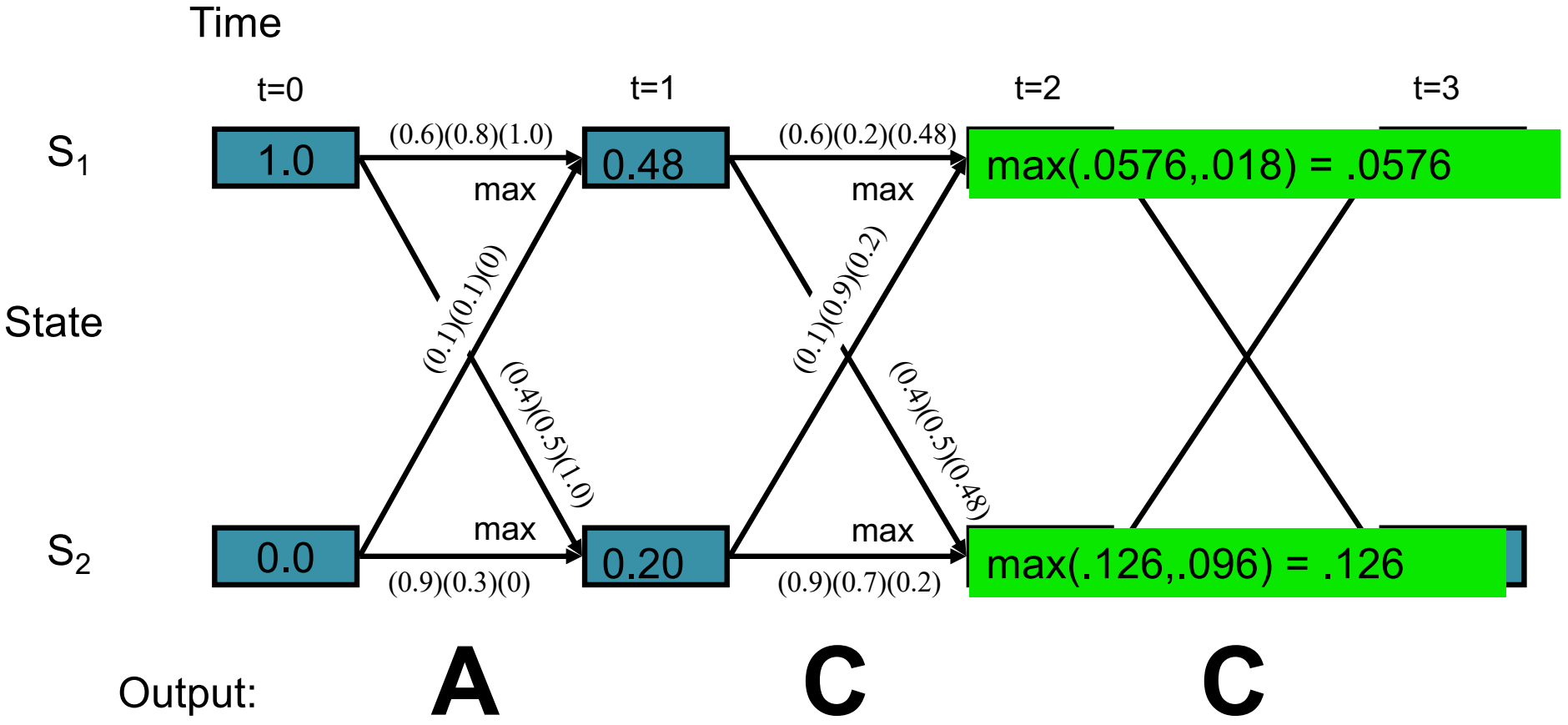
A trellis for the Viterbi Algorithm



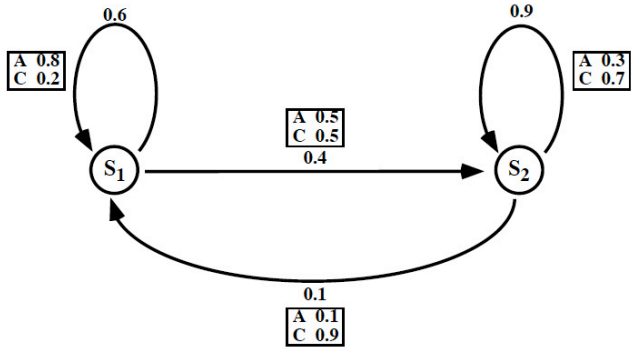


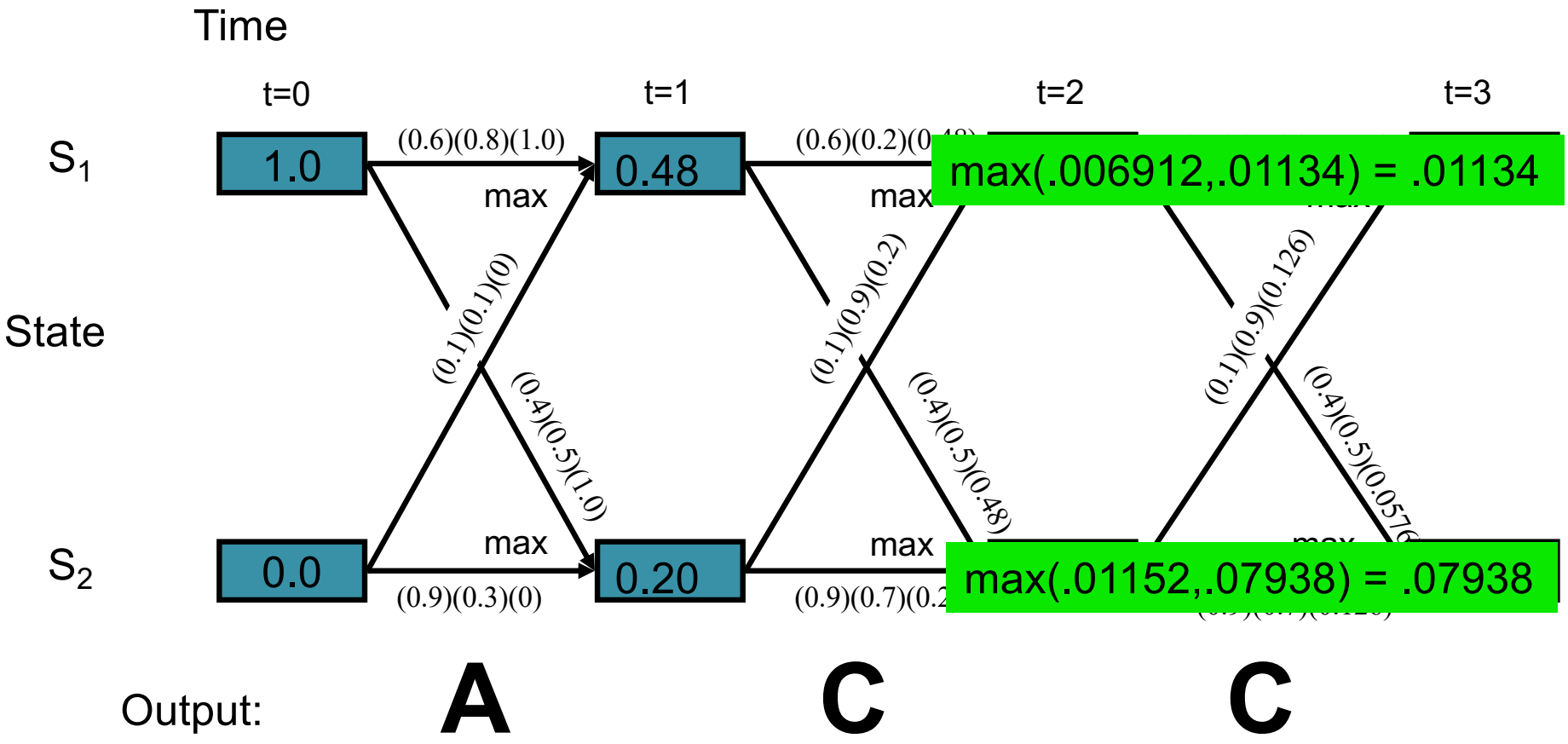
A trellis for the Viterbi Algorithm



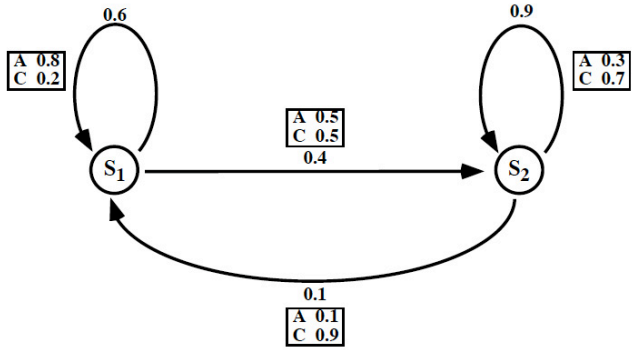


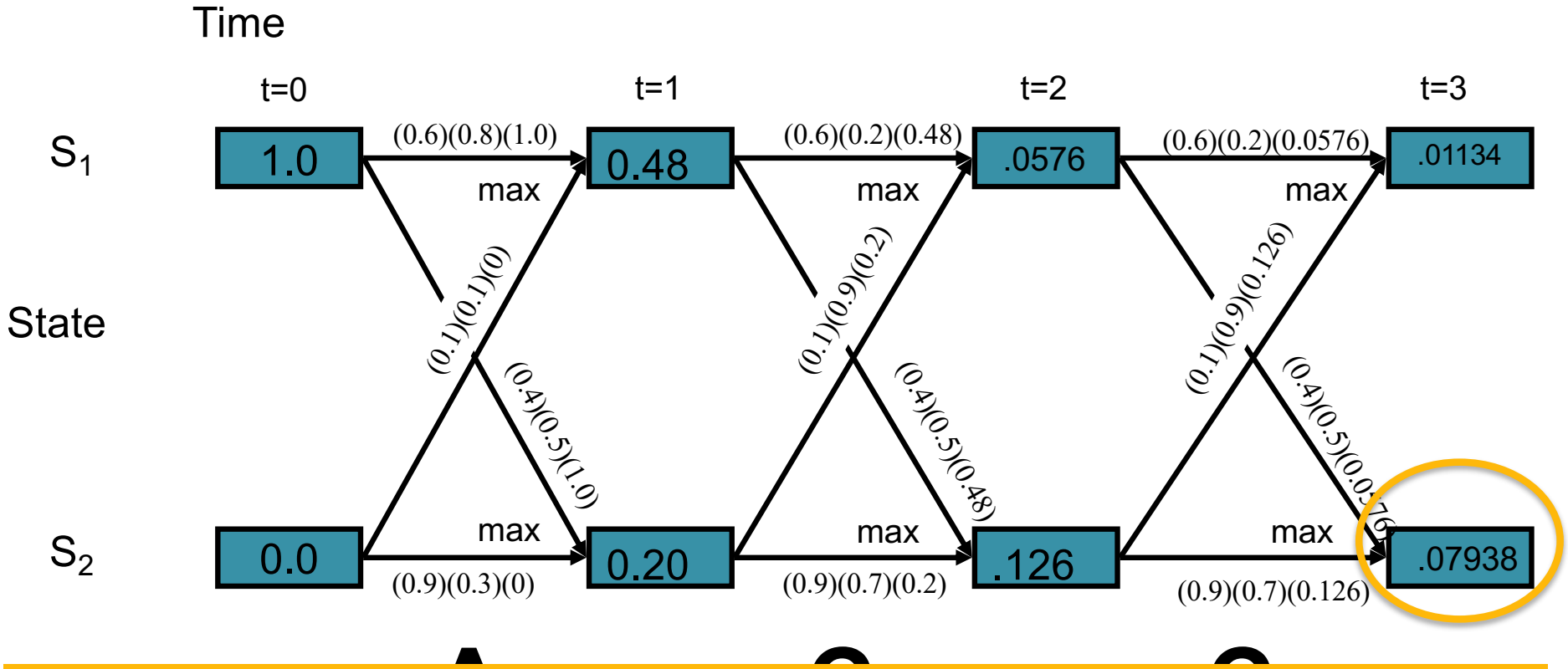
A trellis for the Viterbi Algorithm





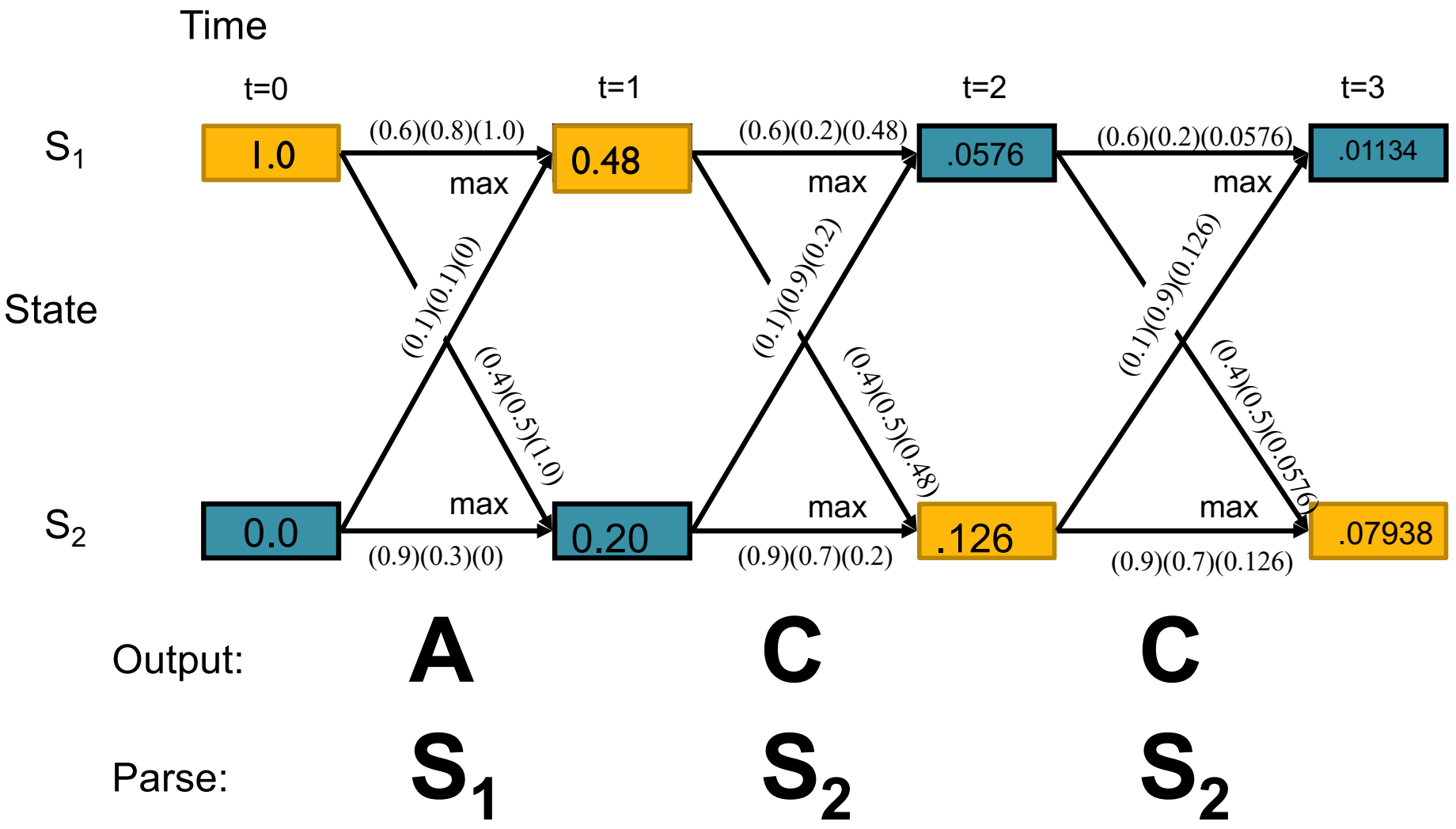
A trellis for the Viterbi Algorithm





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

A trellis for the Viterbi Algorithm



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

Eukaryotic Gene Finding with GlimmerHMM

Mihaela Pertea

Associate Professor

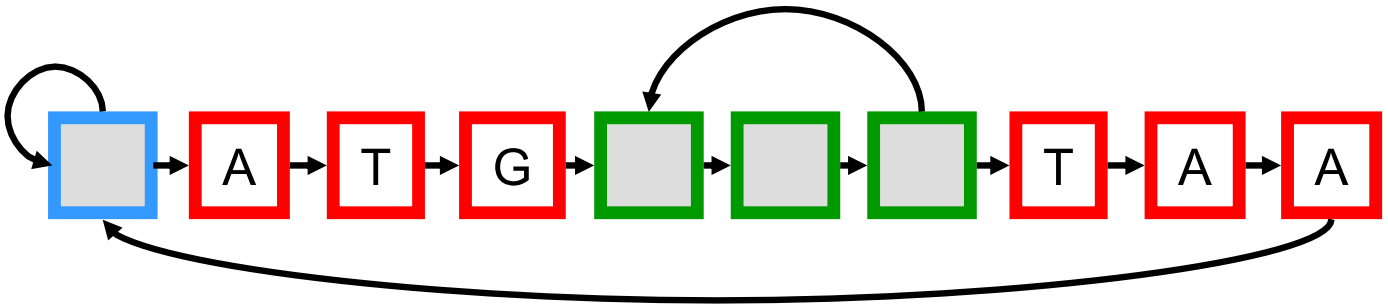
JHU

HMMs and Gene Structure

- Nucleotides {*A,C,G,T*} are the observables

- Different states generate nucleotides at different frequencies

  A simple HMM for unspliced genes:



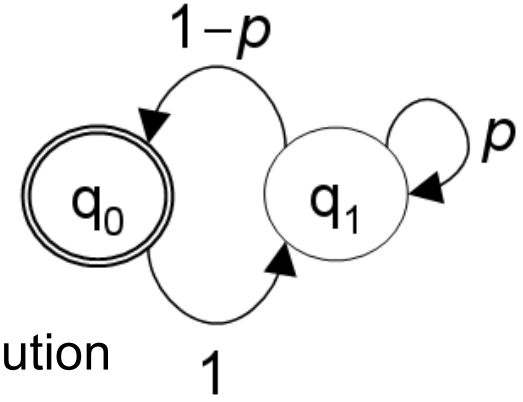
  AAAGC ATG CAT TTA ACG AGA GCA CAA GGG CTC TAA TGCCG

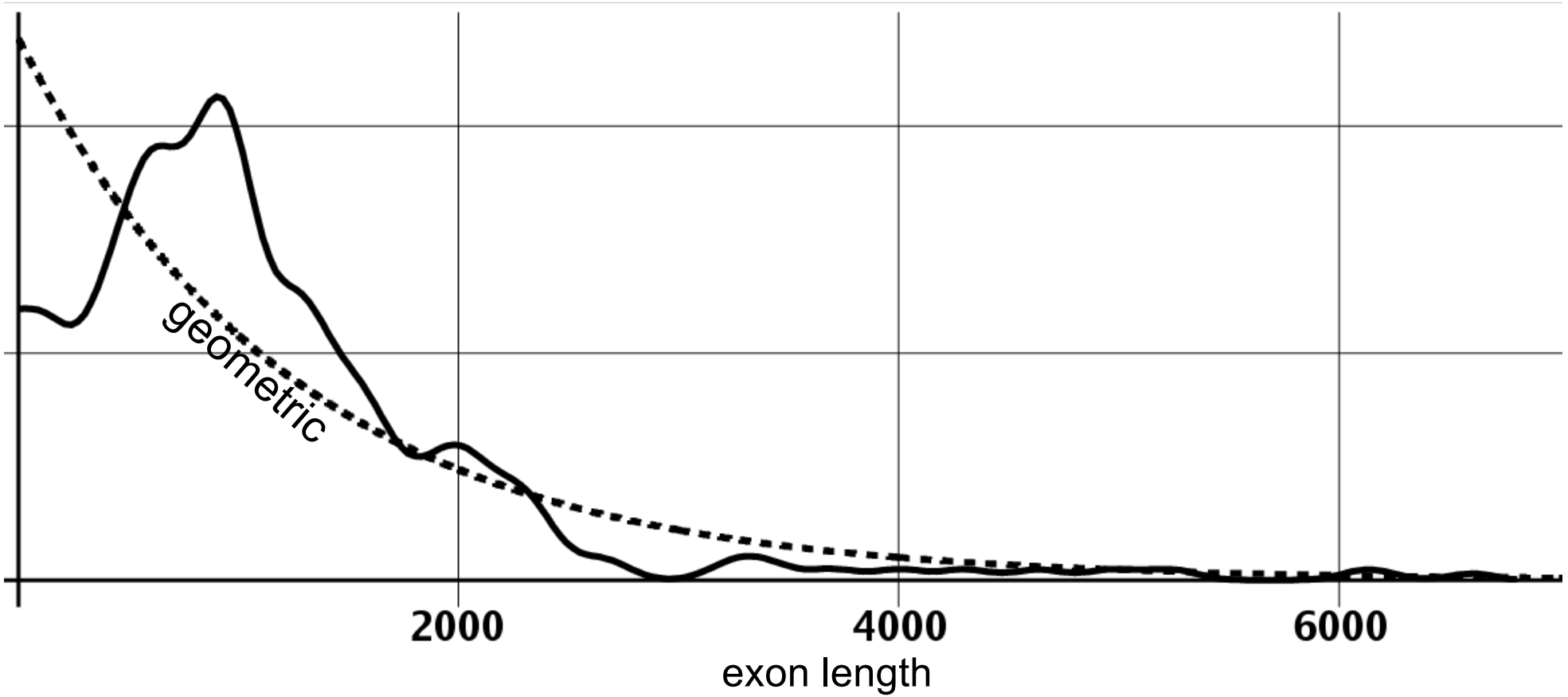
- The sequence of states is an annotation of the generated string – each nucleotide is generated in intergenic, start/stop, coding state

HMMs & Geometric Feature Lengths

$$P(x_0 \ldots x_{d-1} \mid \theta) = \left(\prod_{i=0}^{d-1} P_e(x_i \mid \theta) \right) p^{d-1}(1-p)$$

geometric distribution





Generalized HMMs Summary

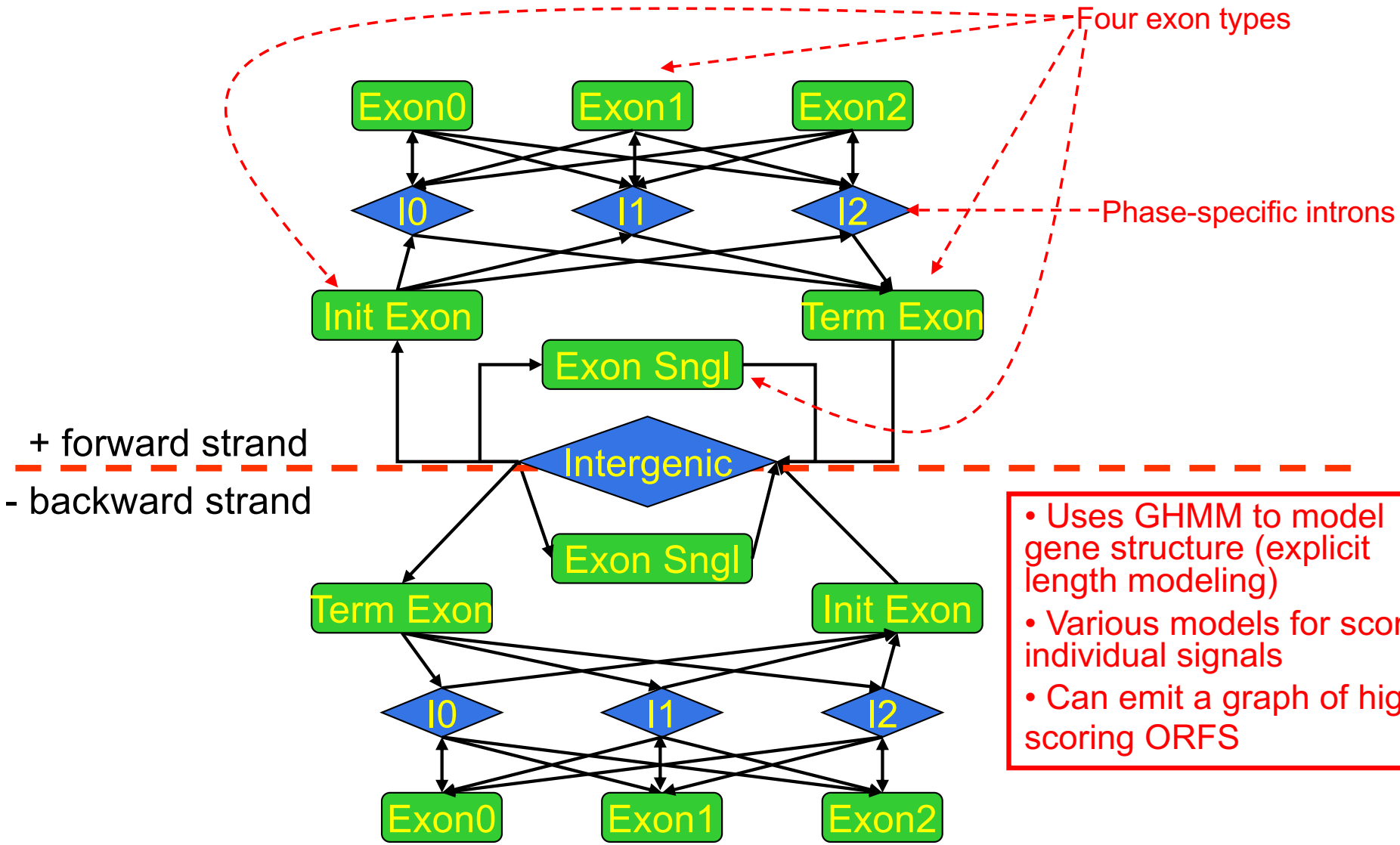
- GHMMs generalize HMMs by allowing each state to emit a subsequence rather than just a single symbol

- Whereas HMMs model all feature lengths using a geometric distribution, coding features can be modeled using an arbitrary length distribution in a GHMM

- Emission models within a GHMM can be any arbitrary probabilistic model ("submodel abstraction"), such as a neural network or decision tree

- GHMMs tend to have many fewer states => simplicity & modularity

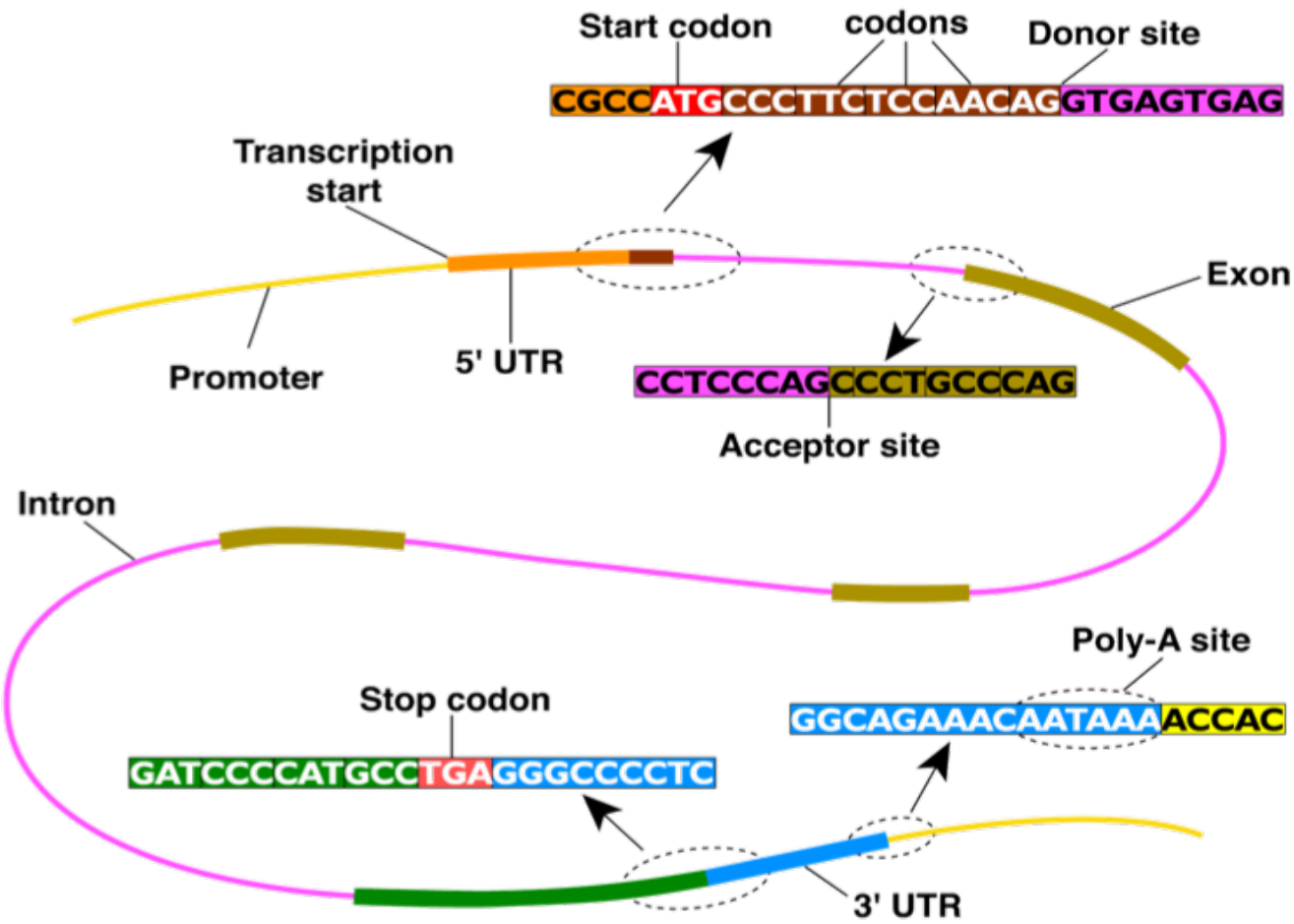
GlimmerHMM architecture



- Uses GHMM to model gene structure (explicit length modeling)
- Various models for scoring individual signals
- Can emit a graph of high-scoring ORFS

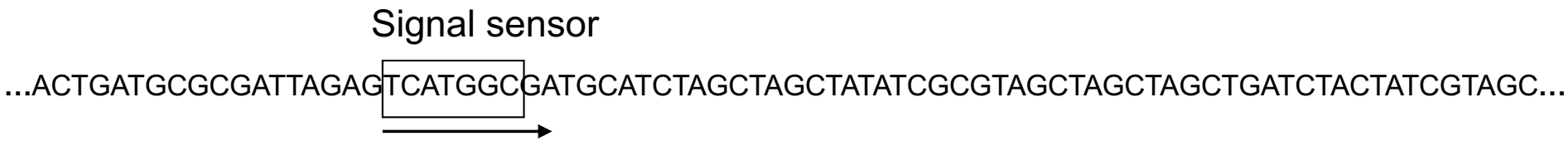
Signal Sensors

Signals – short sequence patterns in the genomic DNA that are recognized by the cellular machinery.



Identifying Signals In DNA

We slide a fixed-length model or "window" along the DNA and evaluate `score(signal)` at each point:

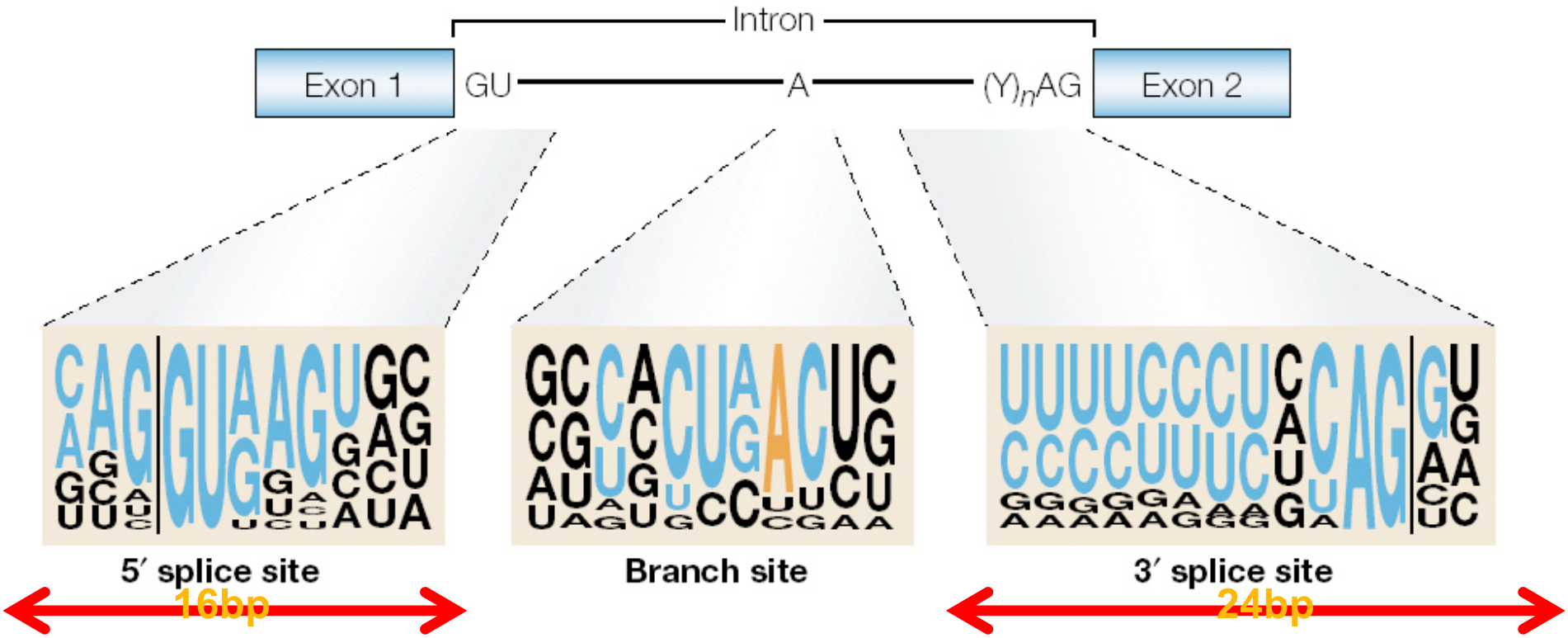


When the `score` is greater than some threshold (determined empirically to result in a desired sensitivity), we remember this position as being the potential site of a signal.

The most common signal sensor is the Position Weight Matrix:

A = 31% T = 28% C = 21% G = 20%	A = 18% T = 32% C = 24% G = 26%	**A** 100%	**T** 100%	**G** 100%	A = 19% T = 20% C = 29% G = 32%	A = 24% T = 18% C = 26% G = 32%

Splice site prediction

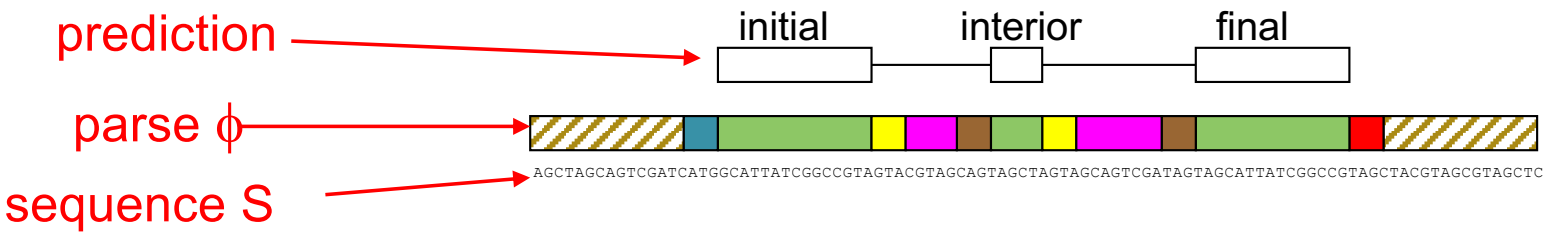


The splice site score is a combination of:

- first or second order inhomogeneous Markov models on windows around the acceptor and donor sites
- Maximal dependence decomposition (MDD) decision trees
- longer Markov models to capture difference between coding and non-coding on opposite sides of site (optional)
- maximal splice site score within 60 bp (optional)

Gene Prediction with a GHMM

Given a sequence S, we would like to determine the parse ϕ of that sequence which segments the DNA into the most likely exon/intron structure:

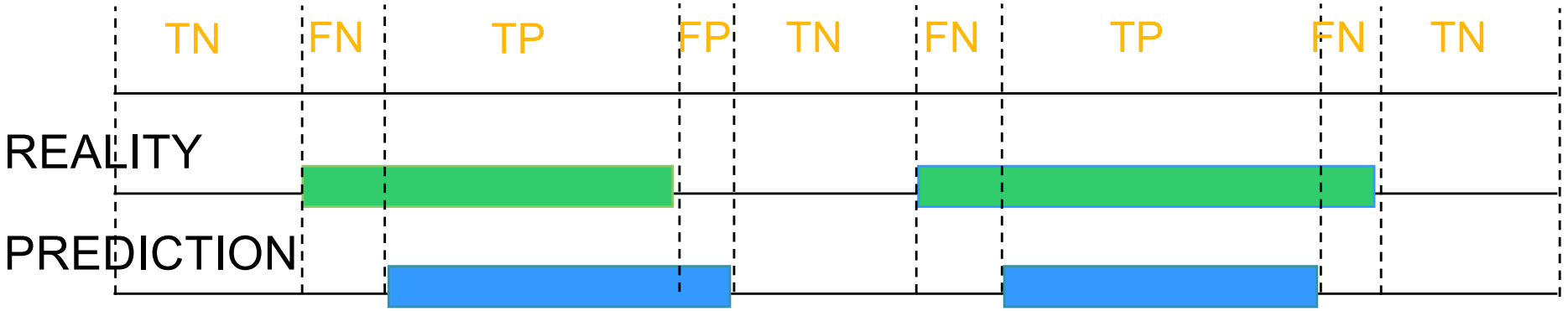


The parse ϕ consists of the coordinates of the predicted exons, and corresponds to the precise sequence of states during the operation of the GHMM (and their duration, which equals the number of symbols each state emits).

This is the same as in an HMM except that in the HMM each state emits bases with fixed probability, whereas in the GHMM each state emits an entire feature such as an exon or intron.

Evaluation of Gene Finding Programs

Nucleotide level accuracy

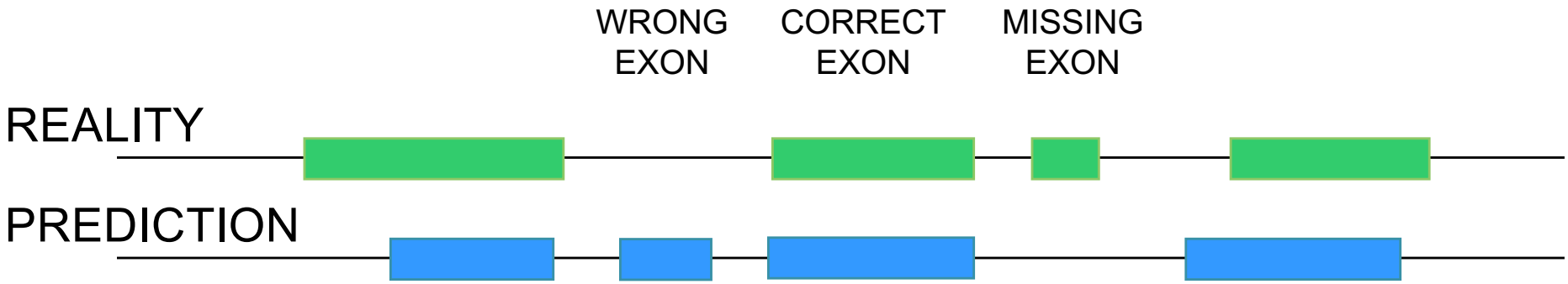


Sensitivity: $Sn = \frac{TP}{TP + FN}$ What fraction of reality did you predict?

Specificity: $Sp = \frac{TP}{TP + FP}$ What fraction of your predictions are real?

More Measures of Prediction Accuracy

Exon level accuracy



$$ExonSn = \frac{TE}{AE} = \frac{\text{number of correct exons}}{\text{number of actual exons}}$$

$$ExonSp = \frac{TE}{PE} = \frac{\text{number of correct exons}}{\text{number of predicted exons}}$$

GlimmerHMM is a high-performance ab initio gene finder

Arabidopsis thaliana test results

	Nucleotide			Exon			Gene		
	Sn	Sp	**Acc**	Sn	Sp	**Acc**	Sn	Sp	**Acc**
GlimmerHMM	97	99	**98**	84	89	**86.5**	60	61	**60.5**
SNAP	96	99	**97.5**	83	85	**84**	60	57	**58.5**
Genscan+	93	99	**96**	74	81	**77.5**	35	35	**35**

- All three programs were tested on a test data set of 809 genes, which did not overlap with the training data set of GlimmerHMM.
- All genes were confirmed by full-length Arabidopsis cDNAs and carefully inspected to remove homologues.

GlimmerHMM on human data

	Nuc Sens	*Nuc Spec*	*Nuc Acc*	*Exon Sens*	*Exon Spec*	*Exon Acc*	*Exact Genes*
GlimmerHMM	86%	72%	79%	72%	62%	67%	17%
Genscan	86%	68%	77%	69%	60%	65%	13%

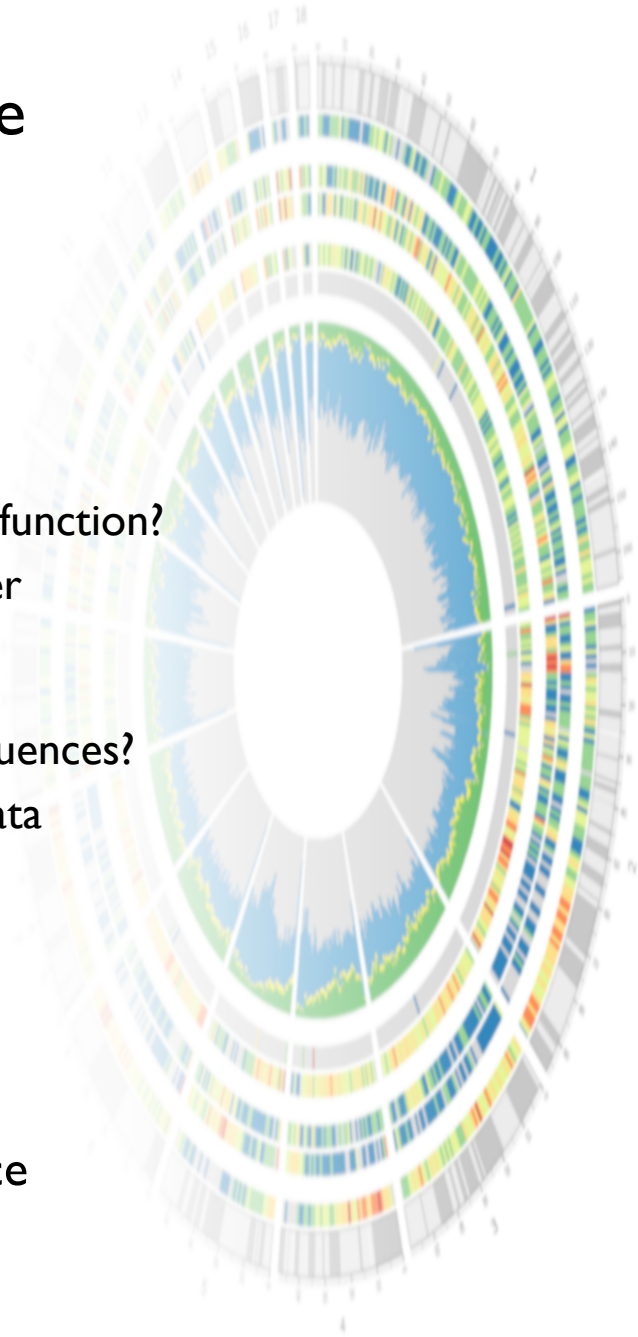
GlimmerHMM's performace compared to Genscan on 963 human RefSeq genes selected randomly from all 24 chromosomes, non-overlapping with the training set. The test set contains 1000 bp of untranslated sequence on either side (5' or 3') of the coding portion of each gene.

Gene Prediction 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

Annotation Summary

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



- 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 ☺