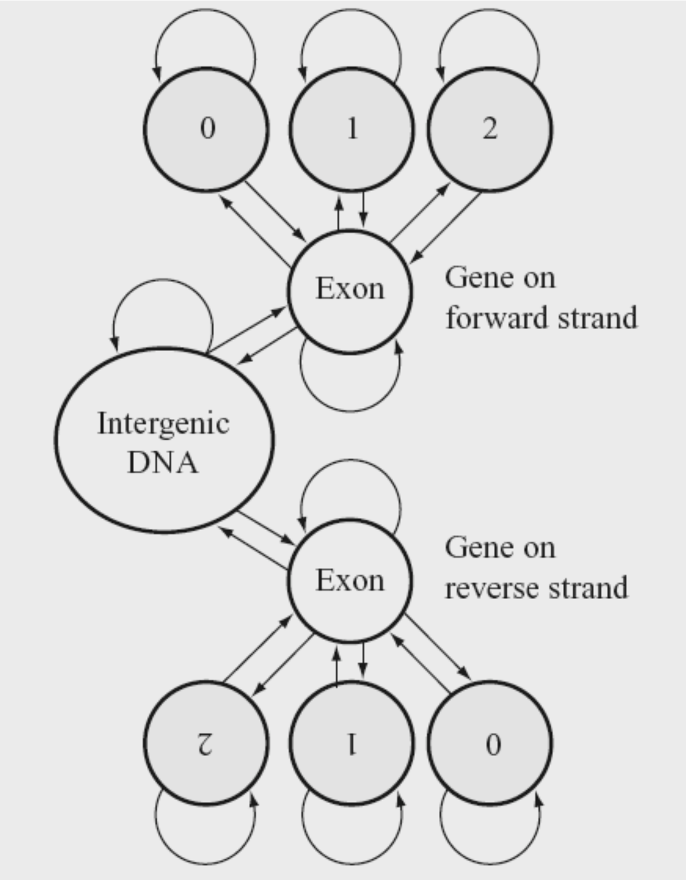
Introduction to Hidden Markov Models for Gene Prediction

ECE-S690



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

Outline

- ✓ Markov Models
- ✓ The Hidden Part
- ✓ How can we use this for gene prediction?

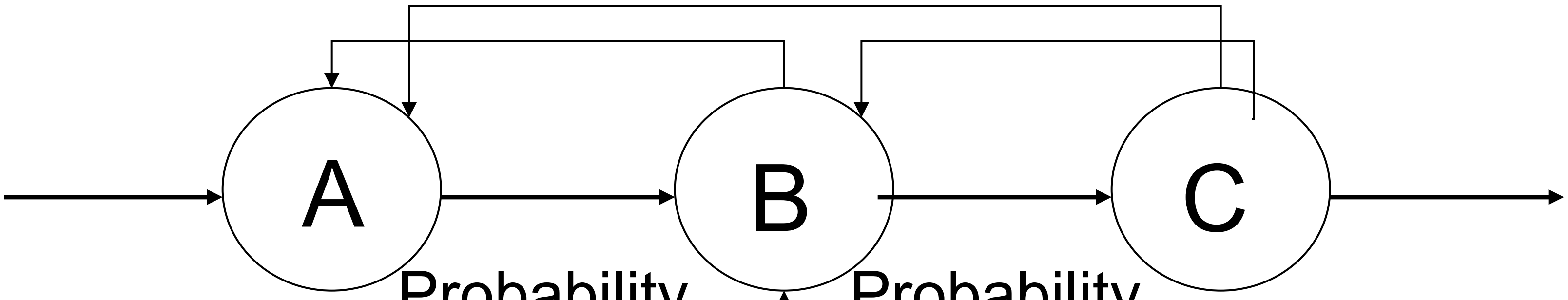
Learning Models

- ✓ Want to recognize patterns (e.g. sequence motifs), we have to learn from the data

- Stochastic process with the Markov Property
  - Stochastic processes are generally looked at as collections of random variables
  - **Markov Property is simply that given the present state, future states are independent of the past.**

- Think of a Markov Chain as a system we can use to predict the future given the present
- Additionally in these systems the present state only depends on two things:
  - Previous state
  - Probability of moving from previous state to present state

Markov Chains



Probability of Transition

Probability of Transition

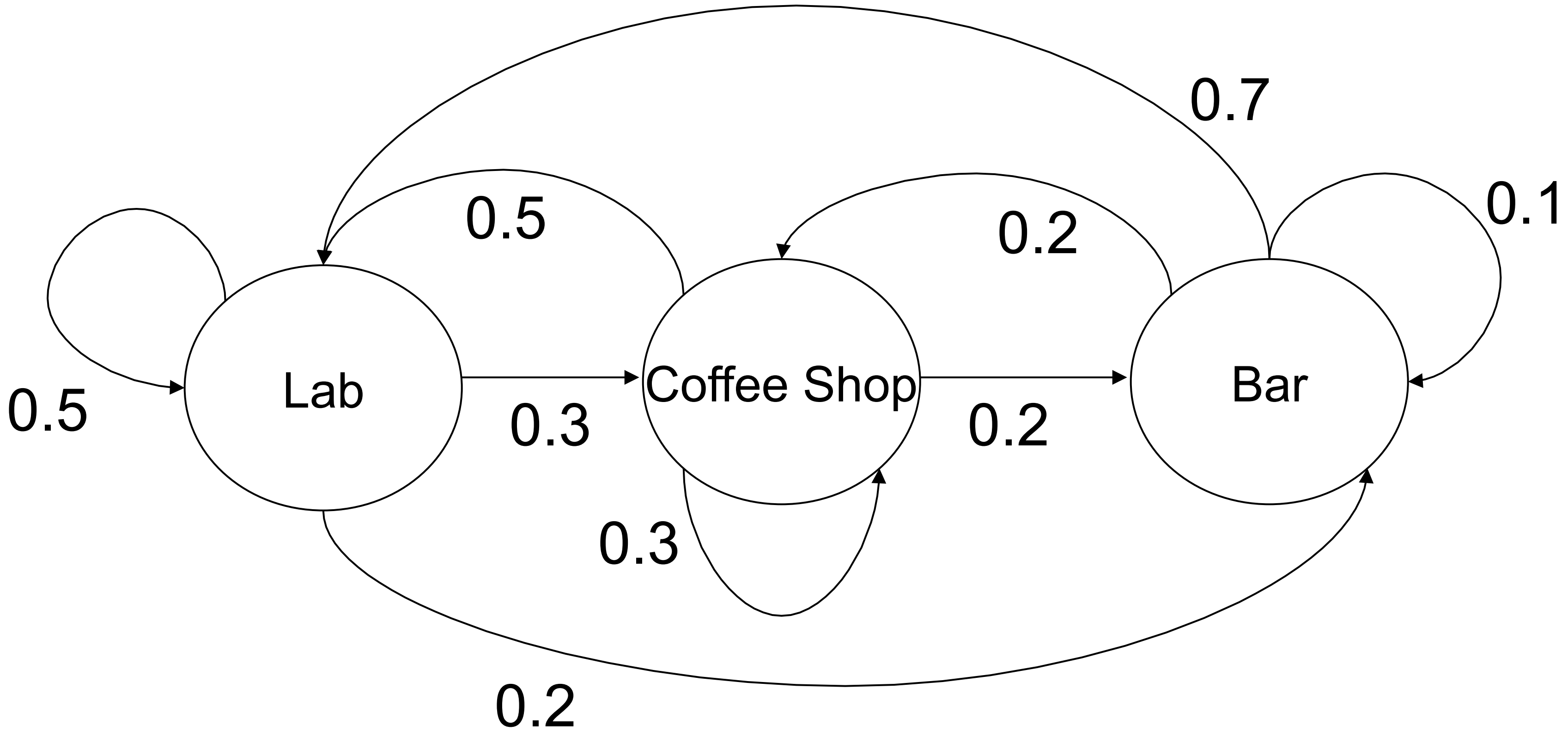
$a_{BA}=Pr(x_i=B|x_{i-1}=A)$

Current State only depends on previous state and transition probability

Example: Estimating Mood State from Grad Student Observations

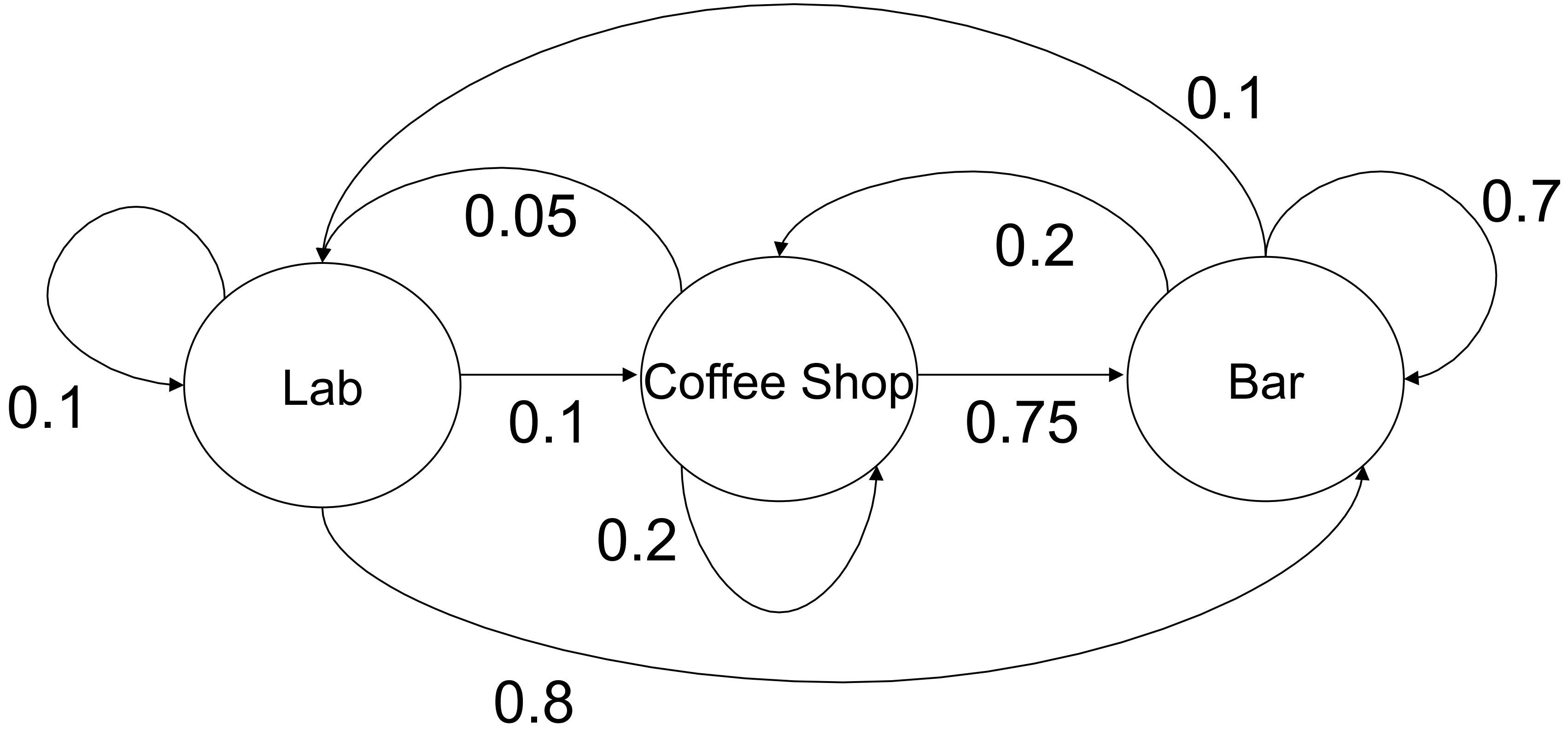
- Grad Student come in two flavors:
  - Happy
  - Depressed about research
- Each type of grad student has it's own Markov chain associated with it.
- Finally, there are three locations we can observe the grad students at:
  - Lab
  - Coffee Shop
  - Bar

Example: “Happy” Grad Student Markov Chain



Observations:
Lab, Coffee, Lab, Coffee, Lab, Lab, Bar, Lab, Coffee,…

Depressed about research



Evaluating Observations

✓ The probability of observing a given sequence is equal to the product of all observed transition probabilities.

$$\Pr(x_1)\prod_{i=2}^{L}\Pr(x_i \mid x_{i-1})$$

X are the observations

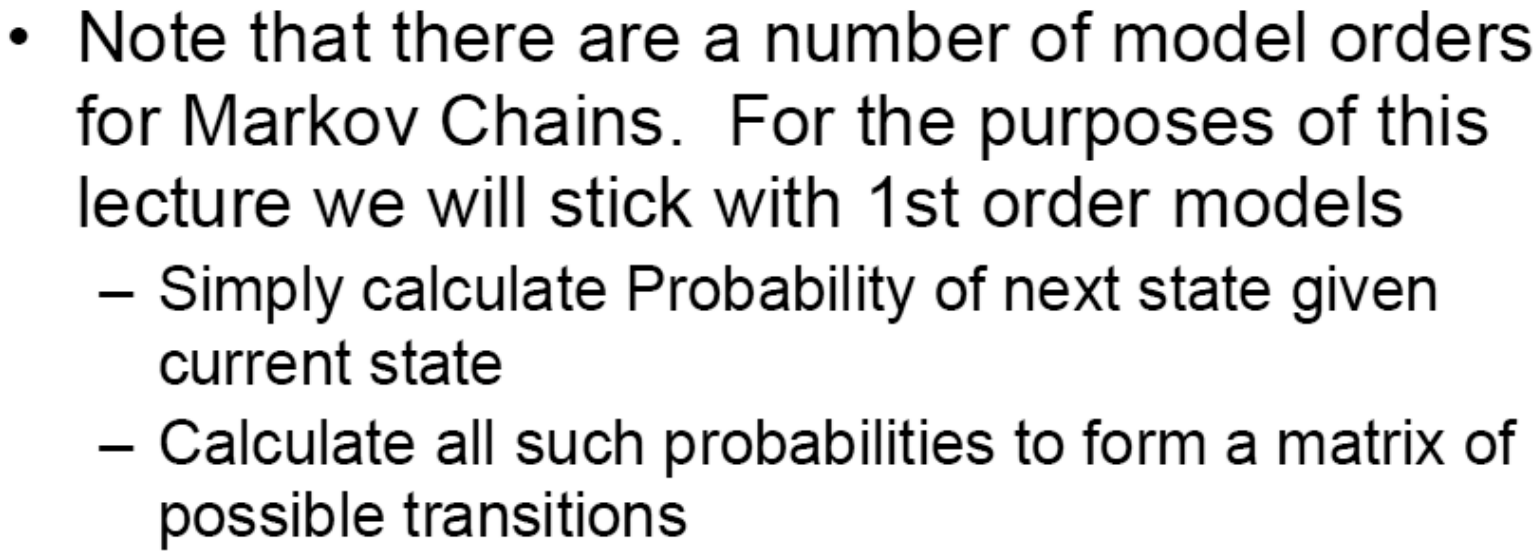
✓ P(Coffee->Bar->Lab) =

P(Coffee) P(Bar | Coffee) P(Lab | Bar)

P(CBL) = P(L|B) P(B|C)P(C)

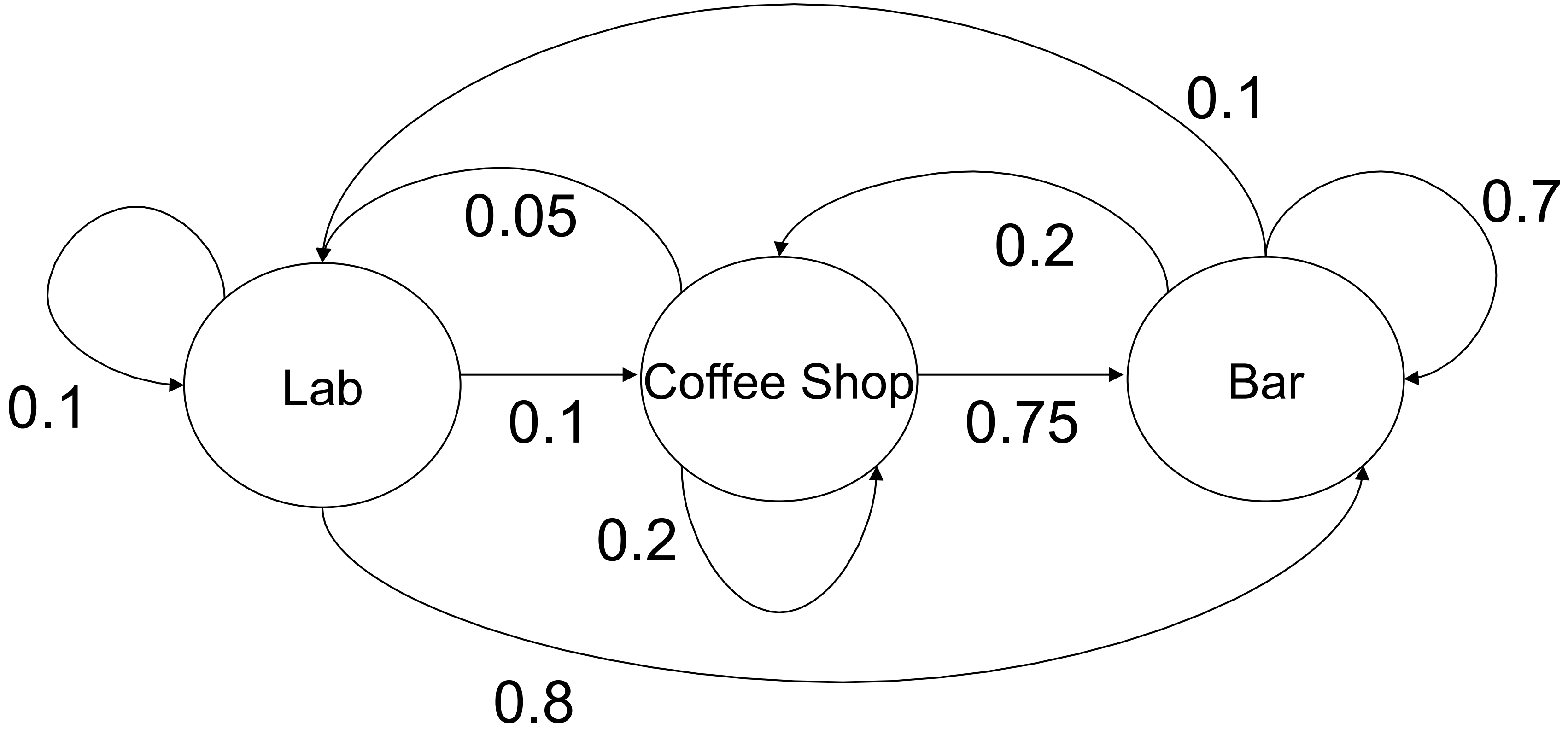
1st order model

- ✓ Probability of Next State | Previous State
  - ✓ Calculate all probabilities



- Note that there are a number of model orders for Markov Chains. For the purposes of this lecture we will stick with 1st order models
  – Simply calculate Probability of next state given current state
  – Calculate all such probabilities to form a matrix of possible transitions

Convert "Depressed" Observations to Matrix



Scoring Observations: Depressed Grad Student

	From Lab	From Coffee Shop	From Bar
To Lab	0.1	0.05	0.2
To Coffee Shop	0.1	0.2	0.1
To Bar	0.8	0.75	0.7

Pr from each state add to 1

Student 1:`LLLCBCLLBBLL`

Student 2:`LCBLBBCBBBBL`

Student 3:`CCLLLLCBCLLL`

Scoring Observations: Depressed Grad Student

	From Lab	From Coffee Shop	From Bar
To Lab	0.1	0.05	0.2
To Coffee Shop	0.1	0.2	0.1
To Bar	0.8	0.75	0.7

Pr from each state add to 1

Student 1: `LLLCBCLLBBLL`

p's

Scoring Observations: Depressed Grad Student

	From Lab	From Coffee Shop	From Bar
To Lab	0.1	0.05	0.2
To Coffee Shop	0.1	0.2	0.1
To Bar	0.8	0.75	0.7

Pr from each state add to 1

Student 1:`LLLCBCLLBBLL`

Student 1:`LLLCBCLLBBLL` = (0.1)(0.1)(0.1)(0.75)(0.1)(0.05)(0.1)(0.8)(0.7)(0.2)(0.1) = 4.2x10-9

Scoring Observations: Depressed Grad Student

	From Lab	From Coffee Shop	From Bar
To Lab	0.1	0.05	0.2
To Coffee Shop	0.1	0.2	0.1
To Bar	0.8	0.75	0.7

Pr from each state add to 1

Student 1:LLLCBCLLBBLL = 4.2x10-9
Student 2:LCBLBBCBBBBL = 4.3x10-5
Student 3:CCLLLLCBCLLL = 3.8x10-11

p’s

Equilibrium State

	From Lab	From Coffee Shop	From Bar
To Lab	0.333	0.333	0.333
To Cofee Shop	0.333	0.333	0.333
To Bar	0.333	0.333	0.333

Student 1:LLLCBCLLBBLL = 5.6x10-6
Student 2:LCBLBBCBBBBL = 5.6x10-6
Student 3:CCCLCCCBCCCL = 5.6x10-6

q's

Comparing to Equilibrium States

$$\frac{\prod_{i} p_{x_i y_i}}{\prod_{i} q_{x_i} q_{y_i}}$$

Likelihood Ratios:

– Simply the ratio of the computed probability of the string of observations given the original chain, divided by the equilibrium.

Evaluation Observations

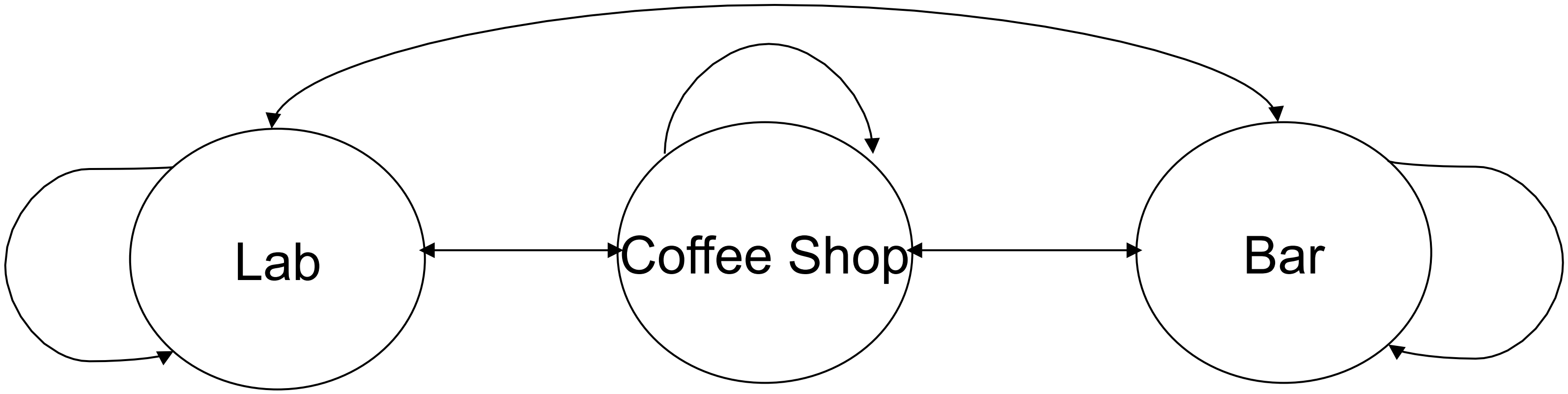
- ✓ Likelihood ratios:
  - ✓ Student 1 = 4.2x10-9 / 5.6x10-6 = 7.5x10-4
  - ✓ Student 2 = 4.3x10-5 / 5.6x10-6 = 7.7
  - ✓ Student 3 = 3.8x10-11 / 5.6x10-6 = 6.8 x 10-6
- ✓ Log likelihood ratios
  - ✓ Student 1 = -3.2
  - ✓ Student 2 = 0.9 (Most likely sad)
  - ✓ Student 3 = -5.2

$$\sum_i \log\left(\frac{p_{x_i y_i}}{q_{x_i} q_{y_i}}\right)$$

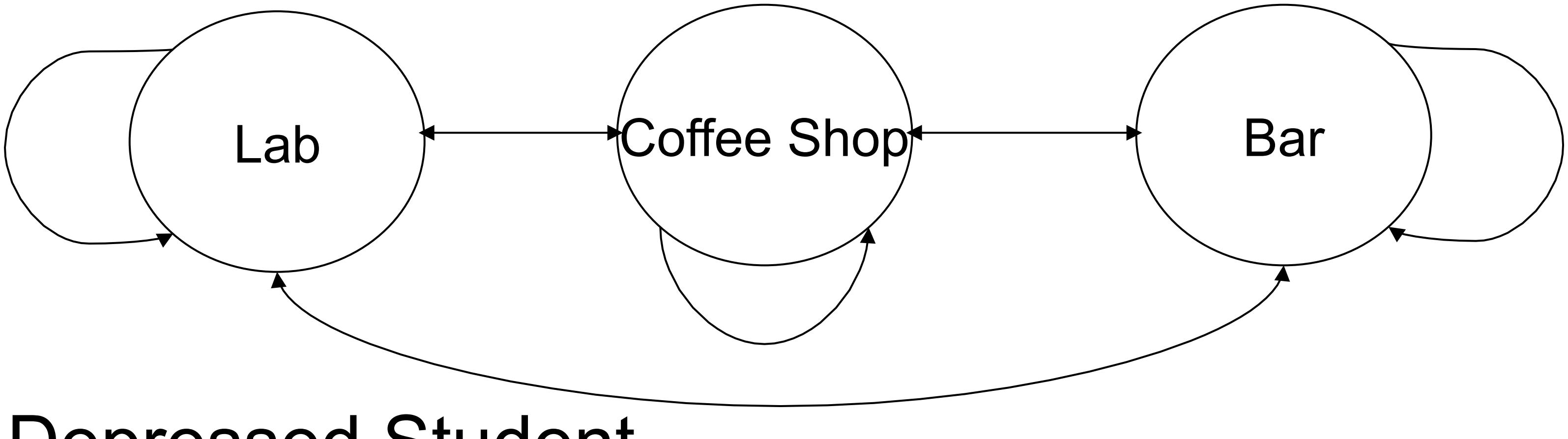
The model could represent Research Breakthrough (Happy) Student!: Transition Probabilities

	From Lab	From Coffee Shop	From Bar
To Lab	0.6	0.75	0.5
To Cofee Shop	0.25	0.2	0.45
To Bar	0.15	0.05	0.05

Combined Model

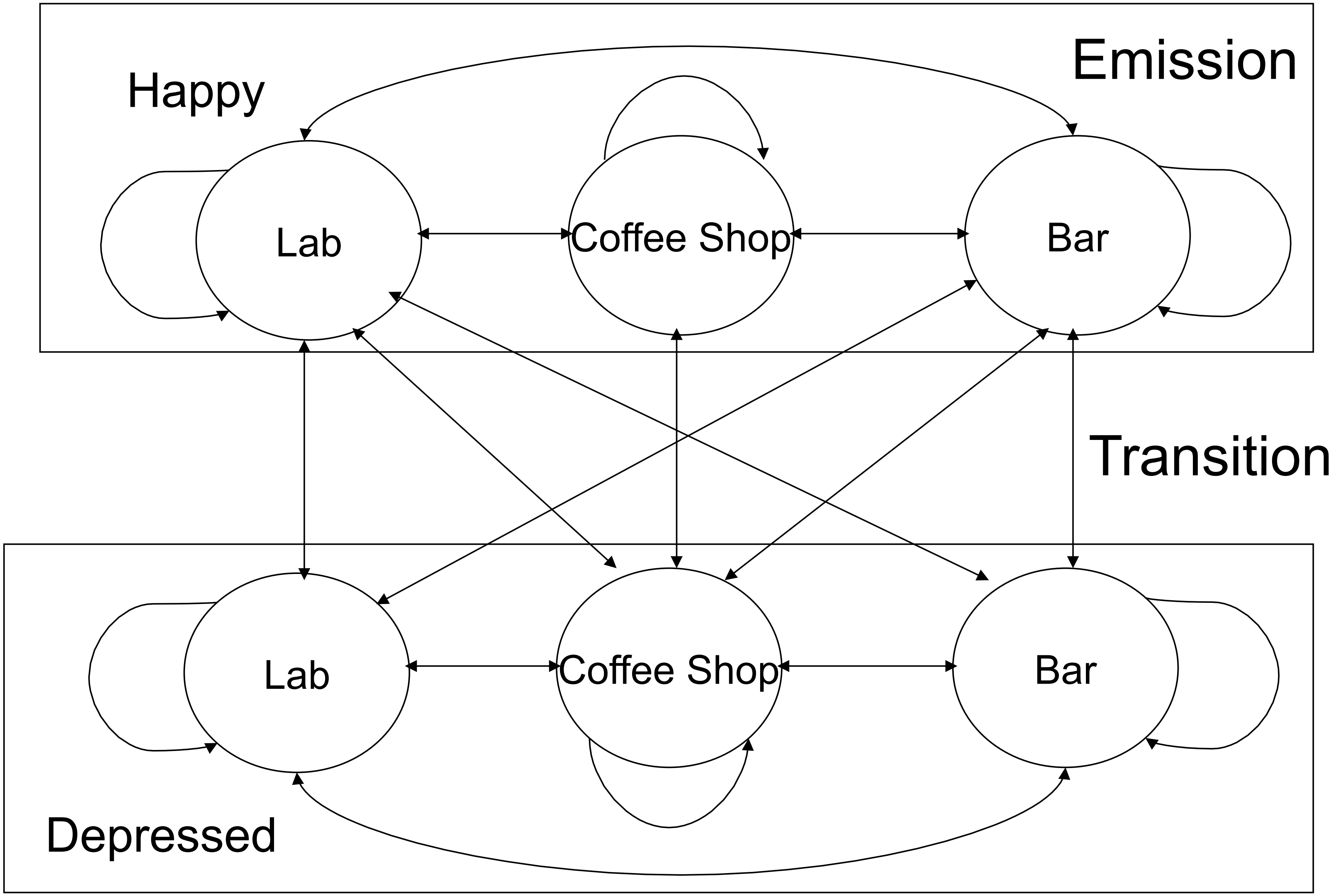


Happy Student

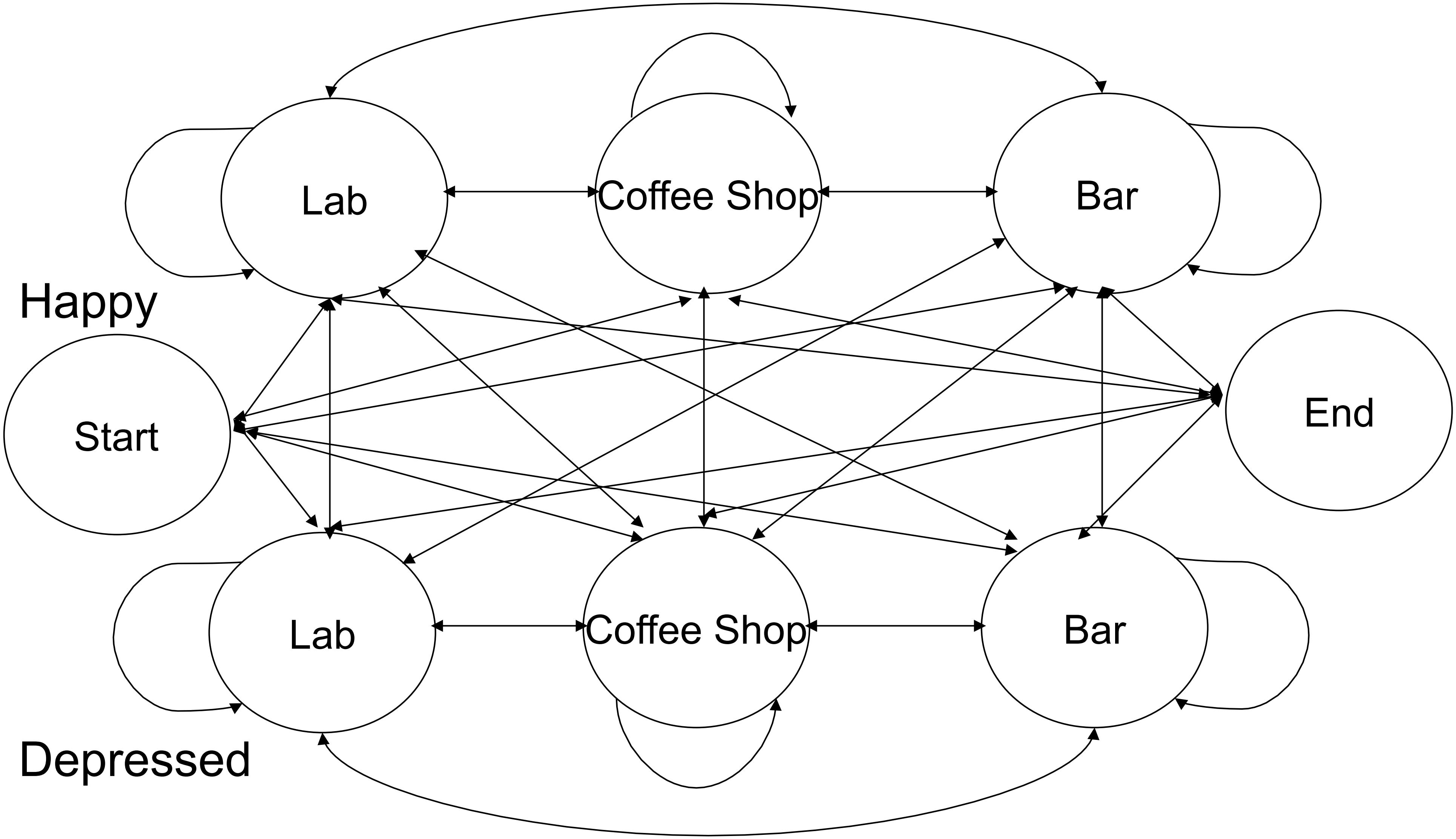


Depressed Student

“Generalized” HMM



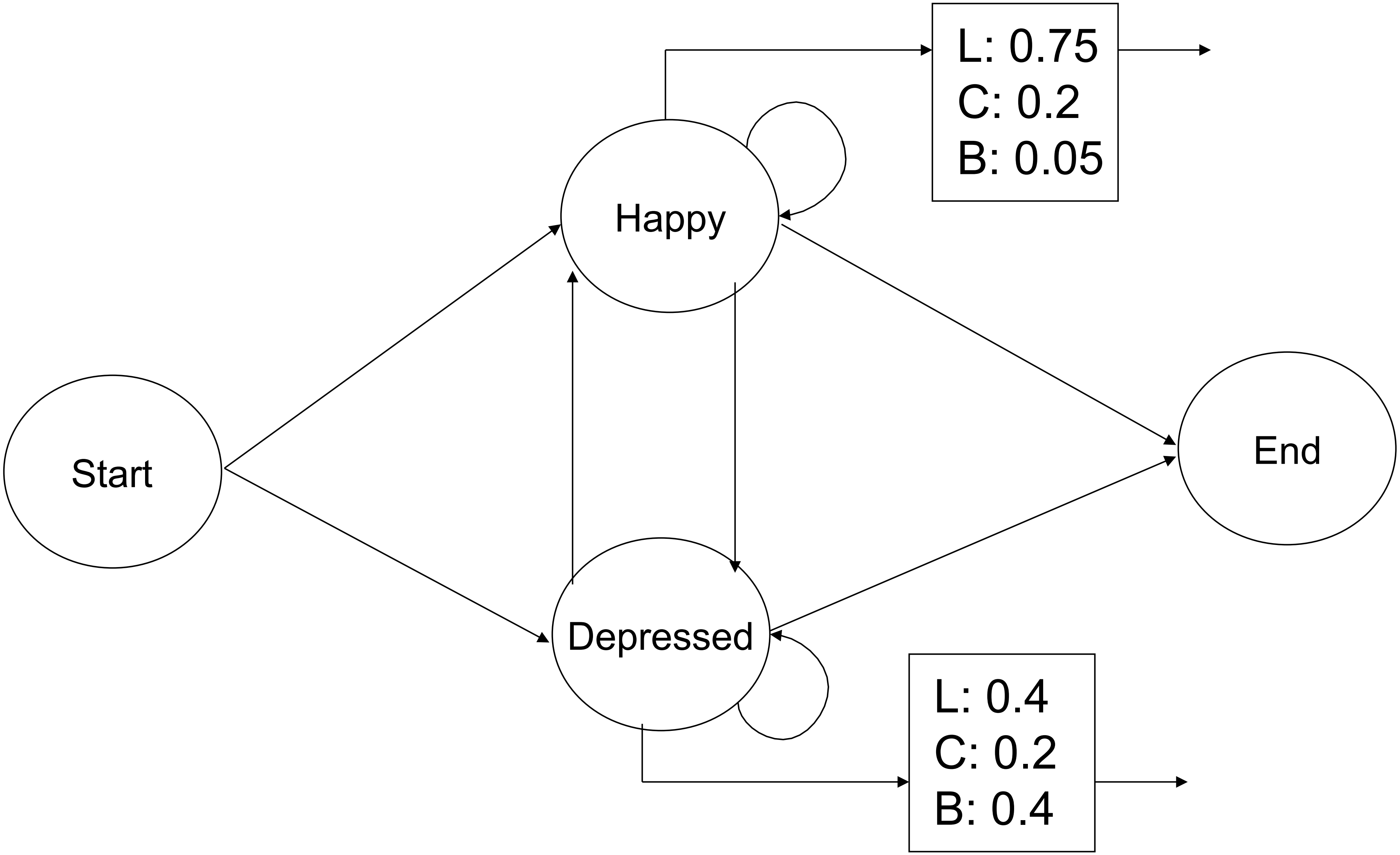
Generalized HMM - Combined Model



Simplifying the Markov Chains to 0^{th} order to model hidden states

- Describe the probability of being in a particular state overall instead of having all the transition probabilities
- **Happy Student:**
  - Lab 75%
  - Coffee 20%
  - Bar 5%
- **Sad Student:**
  - Lab 40%
  - Coffee 20%
  - Bar 40%

HMM - Combined Model



Hiddenness

- Now we have general information about the relationship between state and location
- If we simply observe the locations of the student can we tell what mood they are in?
  - Mood is Hidden
  - Observations are the locations of the students
  - Parameters of the model are the probabilities of a student being in a particular location

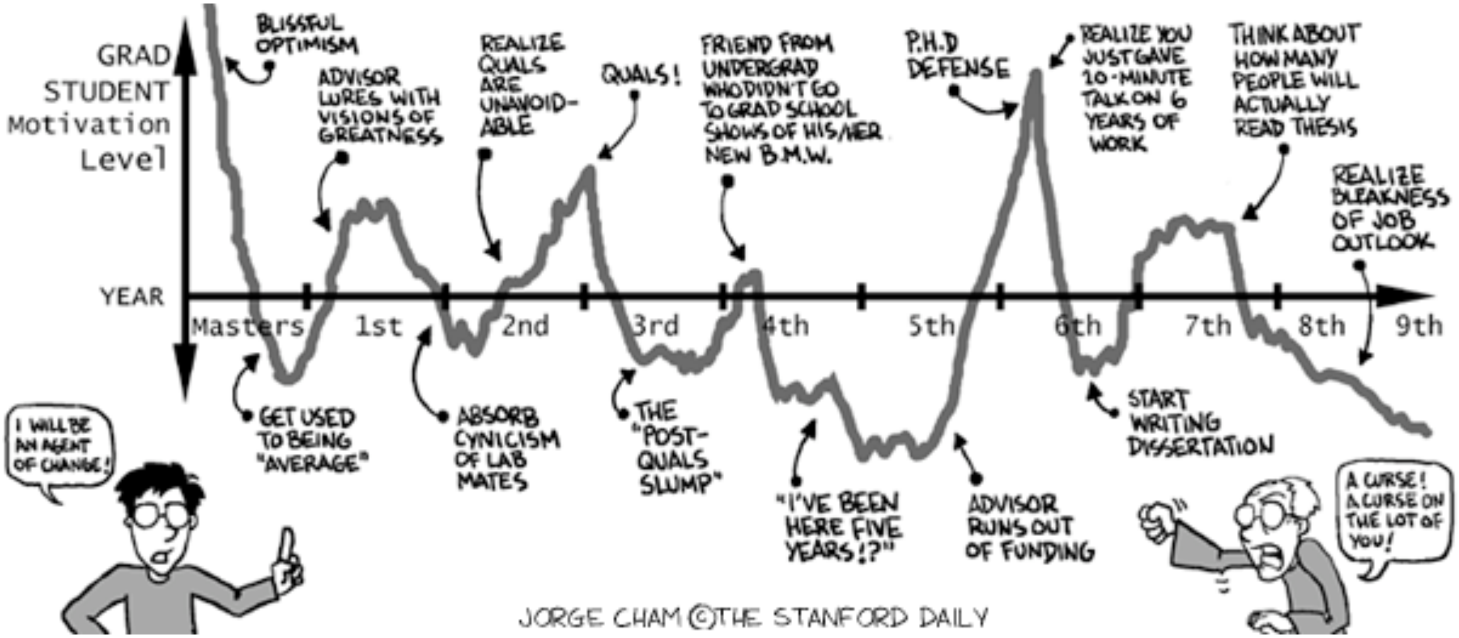
Evaluating Hidden State

- ✓ Evaluating Hidden State
  - ✓ Observations:

    LLLCBCLLBBLLCBLBBCBBBBLCLLLCCL

    Hidden state:

    HHHHHHHHHHHHHDDDDDDDDDDHHHHHHH


GRAD STUDENT Motivation Level
YEAR
Masters
1st
2nd
3rd
4th
5th
6th
7th
8th
9th
BLISSFUL OPTIMISM
ADVISOR LURES WITH VISIONS OF GREATNESS
REALIZE QUALS ARE UNAVOID-ABLE
QUALS!
FRIEND FROM UNDERGRAD WHO DIDN'T GO TO GRAD SCHOOL SHOWS OF HIS/HER NEW B.M.W.
P.H.D DEFENSE
REALIZE YOU JUST GAVE 10-MINUTE TALK ON 6 YEARS OF WORK
THINK ABOUT HOW MANY PEOPLE WILL ACTUALLY READ THESIS
REALIZE BLEAKNESS OF JOB OUTLOOK
I WILL BE AN AGENT OF CHANGE!
GET USED TO BEING "AVERAGE"
ABSORB CYNICISM OF LAB MATES
THE "POST-QUALS SLUMP"
"I'VE BEEN HERE FIVE YEARS!?"
ADVISOR RUNS OUT OF FUNDING
START WRITING DISSERTATION
A CURSE! A CURSE ON THE LOT OF YOU!
JORGE CHAM ©THE STANFORD DAILY

Applications

- Cryptanalysis
  - The study of obtaining encrypted information without access to the secret information which is required to decode it.
- Speech Recognition
  - Identify the person who is speaking knowing only what is being said and a model for probable speakers
- Machine Translation
  - Use computers to translate from one language to another
- Gene Prediction
  - Predicting when a gene is present based on nucleotide observations

Particulars about HMMs

- HMMs ultimately need to be trained to be truly effective
- Give the system a series of observations and allow the model to adjust it's parameters accordingly
- In the gene finding example we feed the system a series of nucleotide sequences that are known to be genes and non genes.

Gene Prediction

- What we want:
  - Find coding and noncoding regions of an unlabeled string of DNA nucleotides
- What's the motivation:
  - Annotate genomic data which is becoming abundant due to next generation sequencing methods
  - Gain insights into the mechanisms involved in transcription, splicing and other processes

Why are HMMs a good fit for DNA and Amino Acids?

- DNA sequences are in a particular order which is necessary for HMMs (can’t have unordered data)
- Lots of training data is available for us to train the system on what is a gene and what is not a gene

HMM Caveats

- States are supposed to be independent of each other and this isn't always true
- Need to be mindful of overfitting

– Need a good training set

– More training data does not always mean a better model

- HMMs can be slow (if proper Decoding not implemented)

– Some decoding maps out all paths through the model

– DNA sequences can be very long so processing/ annotating them can be very time consuming

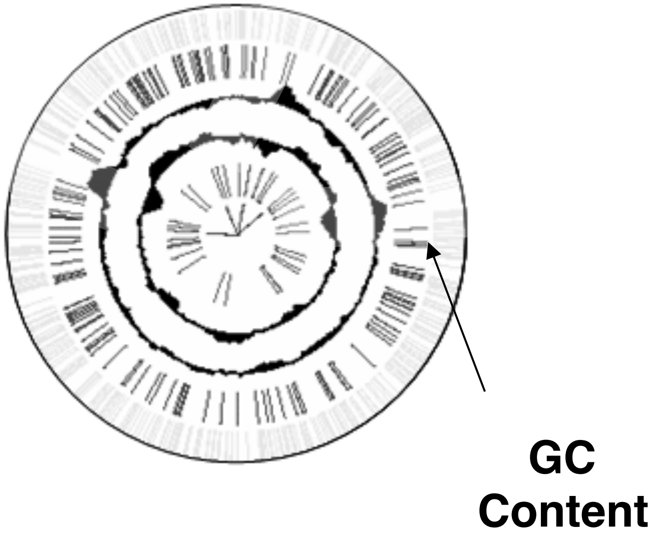
Genomic Applications

- ✓ Finding Genes
- ✓ Finding Pathogenicity Islands

Example Bio App: Pathogenicity Islands

- ✓ Clusters of genes acquired by horizontal transfer
  - ✓ Present in pathogenic species but not others
- ✓ Frequently encode virulence factors
  - ✓ Toxins, secondary metabolites, adhesins

***Neisseria meningitidis*, 52% G+C**



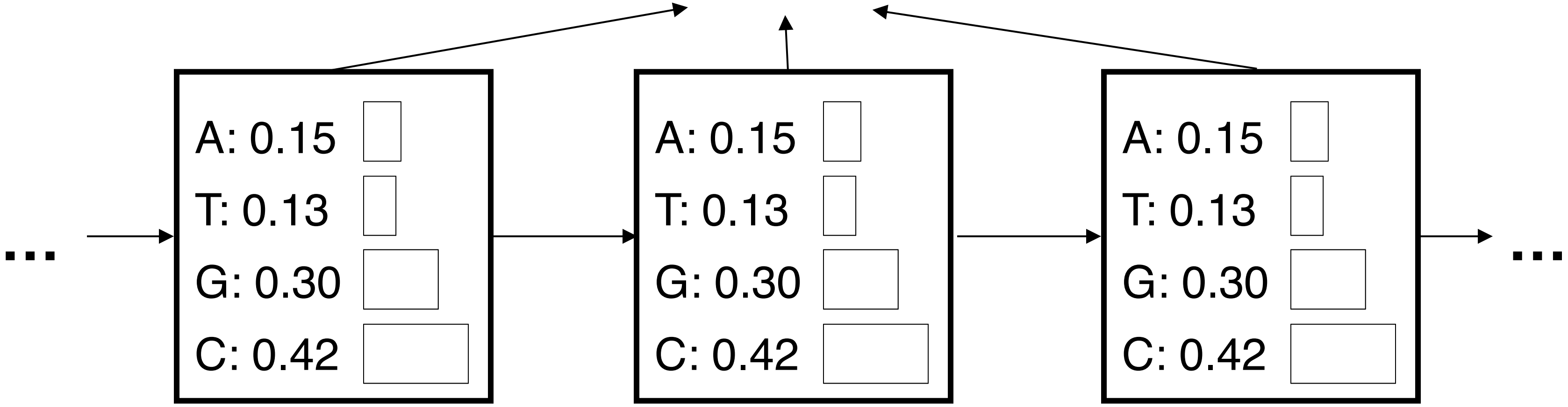
(*from Tettelin et al. 2000. Science*)

- ✓ (Flanked by repeats, regulation and have different codon usage)
- ✓ *Different GC content than rest of genome*

Modeling Sequence Composition (Simple Probability of Sequence)

- ✓ Calculate sequence distribution from known islands
  - ✓ Count occurrences of A,T,G,C
- ✓ Model islands as nucleotides drawn independently from this distribution

... C C TA A G T T A G A G G A T T G A G A ....



...

A: 0.15
T: 0.13
G: 0.30
C: 0.42

A: 0.15
T: 0.13
G: 0.30
C: 0.42

A: 0.15
T: 0.13
G: 0.30
C: 0.42

...

$\mathbf{P(S_i|MP)}$

The Probability of a Sequence (Simplistic)

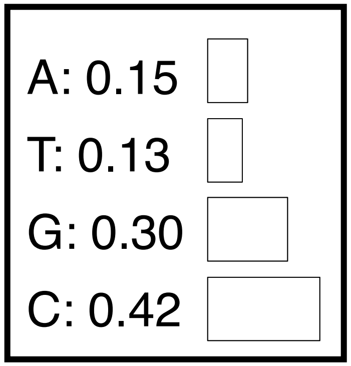
- ✓ Can calculate the probability of a particular sequence (S) according to the pathogenicity island model (MP)

$$P(S \mid MP) = P(S_1, S_2, \ldots S_N \mid MP) = \prod_{i=1}^{N} P(S_i \mid MP)$$

Example

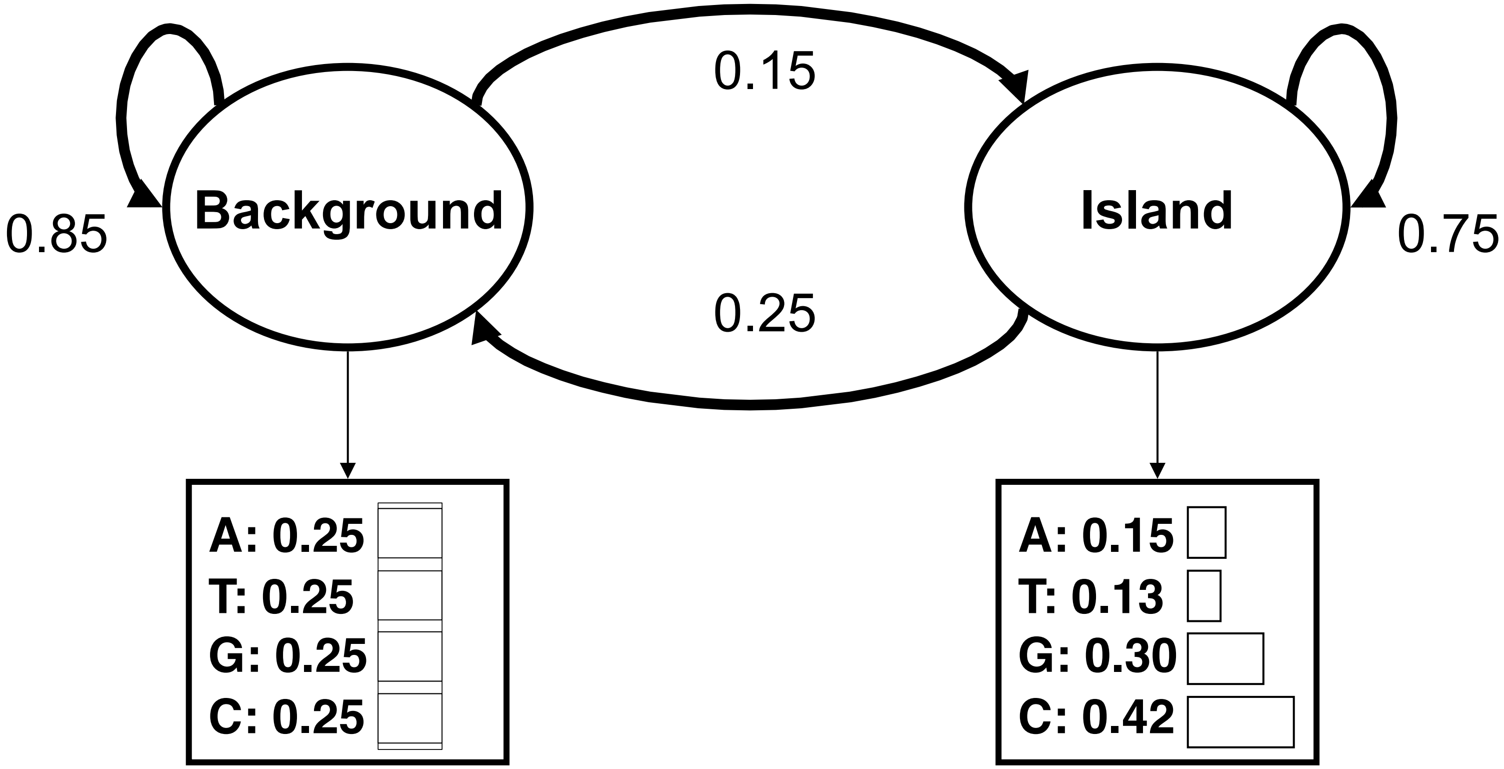
S = AAATGCGCATTTCGAA

$$\begin{aligned} P(S \mid MP) &= P(A)^6 \times P(T)^4 \times P(G)^3 \times P(C)^2 \\ &= (0.15)^6 \times (0.13)^4 \times (0.30)^3 \times (0.42)^2 \\ &= 1.55 \times 10^{-11} \end{aligned}$$



A:	0.15
T:	0.13
G:	0.30
C:	0.42

A More Complex Model



0.15

0.85 Background

Island 0.75

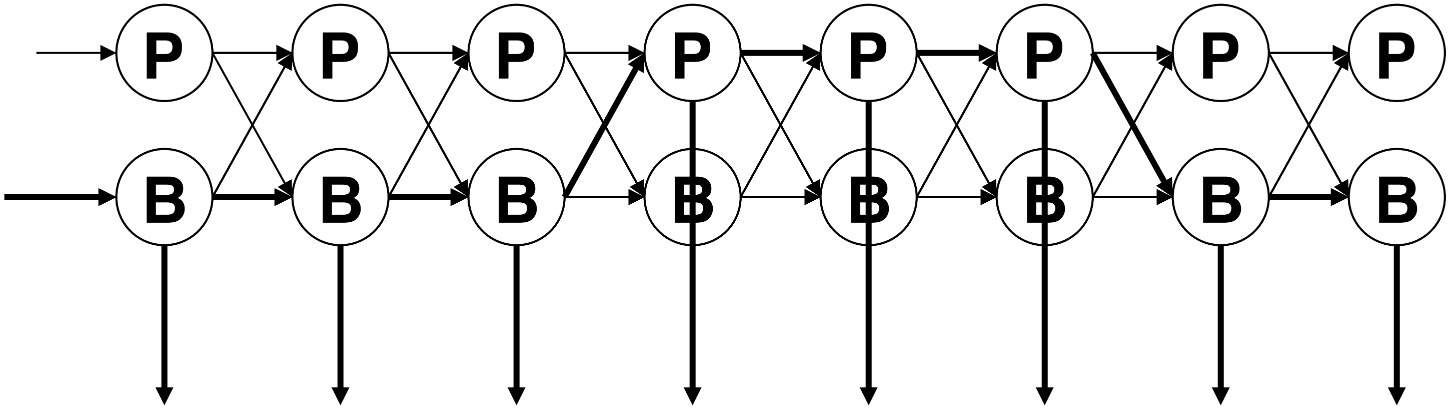
0.25

A: 0.25
T: 0.25
G: 0.25
C: 0.25

A: 0.15
T: 0.13
G: 0.30
C: 0.42

TAAGAATTGTGTCACACACATAAAAACCCTAAGTTAGAGGATTGAGATTGGCA
GACGATTGTTCGTGATAATAAACAAGGGGGCATAGATCAGGCTCATATTGGC

A *Generative* Model



S: G C A A A T G C

$P(L_{i+1}|L_i)$

	B_{i+1}	P_{i+1}
B_i	0.85	0.15
P_i	0.25	0.75

$P(S|B)$

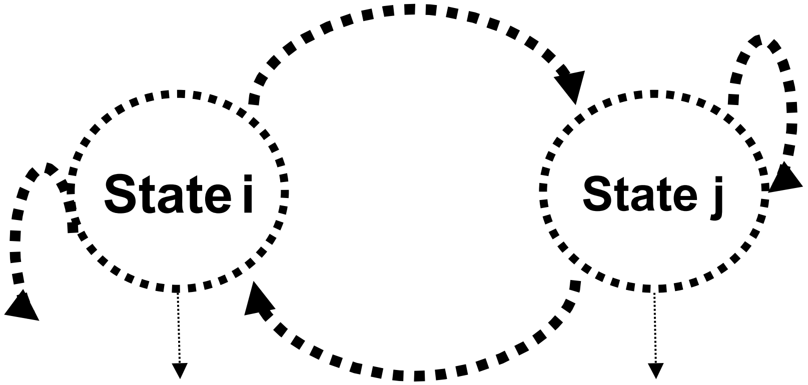
A:	0.25
T:	0.25
G:	0.25
C:	0.25

$P(S|P)$

A:	0.42
T:	0.30
G:	0.13
C:	0.15

The Hidden in HMM

- ✓ DNA does not come conveniently labeled (i.e. Island, Gene, Promoter)
- ✓ We observe nucleotide sequences
- ✓ The *hidden* in HMM refers to the fact that state labels, L, are not observed
  - ✓ Only observe emissions (e.g. nucleotide sequence in our example)



…A A G T T A G A G…

A Hidden Markov Model

Hidden States

L = { 1, ..., K }

Transition probabilities

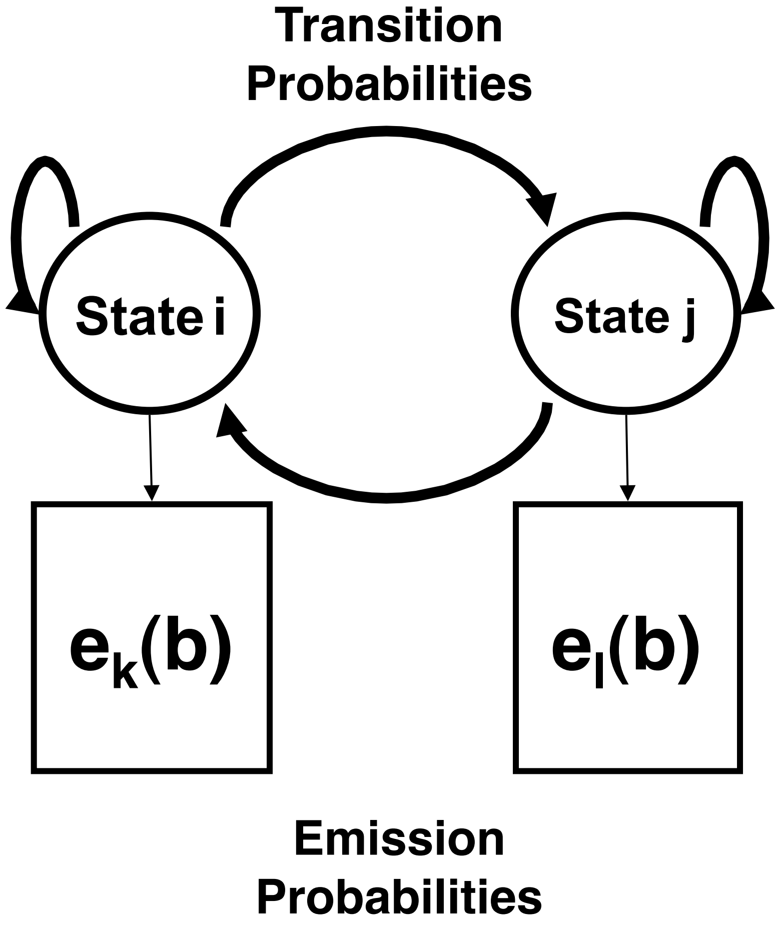
a_{kl} = Transition probability from state k to state l

Emission probabilities

$e_k(b)$ = P(emitting b l state=k)

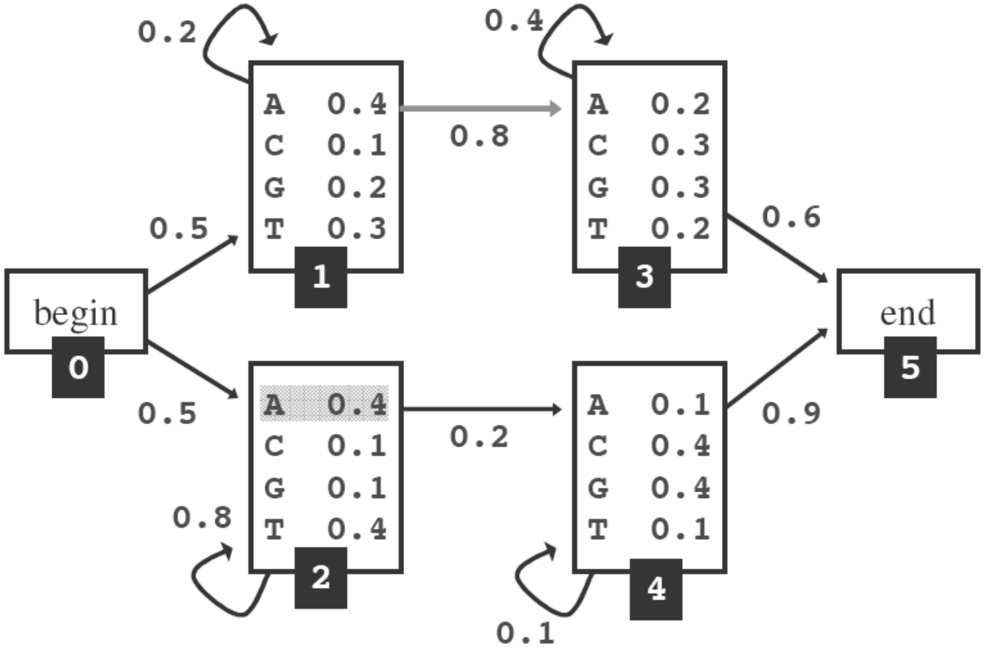
Initial state probability

$\pi(b)$ = P(first state=b)



HMM with Emission Parameters

- ✓ a_{13}: Probability of a transition from State 1 to State 3
- ✓ $e_2(A)$: Probability of emitting character A in state 2



Hidden Markov Models (HMM)

- ✓ Allows you to find sub-sequence that fit your model
- ✓ Hidden states are disconnected from observed states
- ✓ Emission/Transition probabilities
- ✓ Must search for optimal paths

Three Basic Problems of HMMs

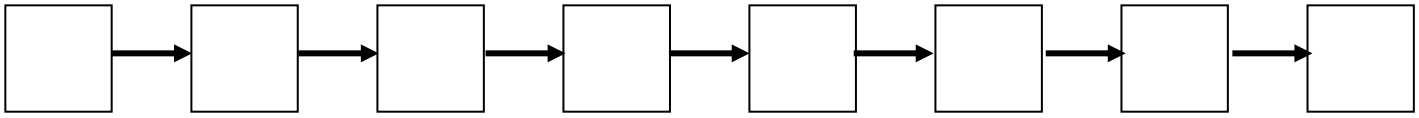
- ✓ The Evaluation Problem
  - ✓ Given an HMM and a sequence of observations, what is the probability that the observations are generated by the model?
- ✓ The Decoding Problem
  - ✓ Given a model and a sequence of observations, what is the most likely state sequence in the model that produced the observations?
- ✓ The Learning Problem
  - ✓ Given a model and a sequence of observations, how should we adjust the model parameters in order to maximize evaluation/decoding

Fundamental HMM Operations

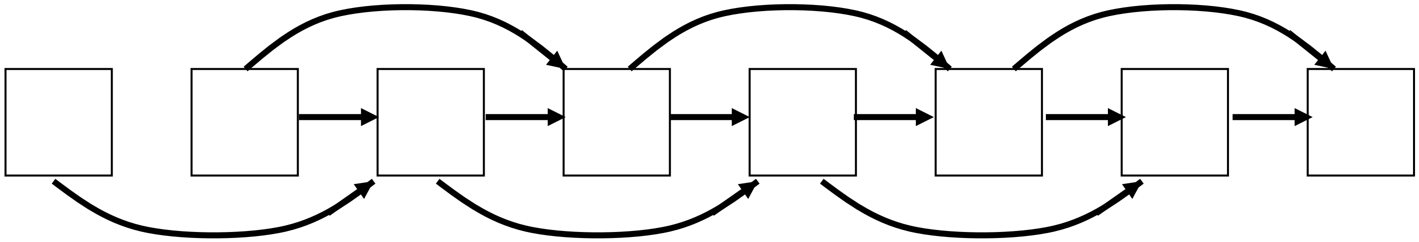
Computation	**Biology**
Decoding ✓ *Given* an HMM and sequence S ✓ *Find of* a corresponding sequence labels, L	**Annotate pathogenicity islands on a new sequence**
Evaluation ✓ *Given* an HMM and sequence S ✓ *Find* P(S|HMM)	**Score a particular sequence**
Training ✓ *Given* an HMM w/o parameters and set of sequences S ✓ *Find* transition and emission probabilities the maximize P(S | params, HMM)	**Learn a model for sequence composed of background DNA and pathogenicity islands**

Markov chains and processes

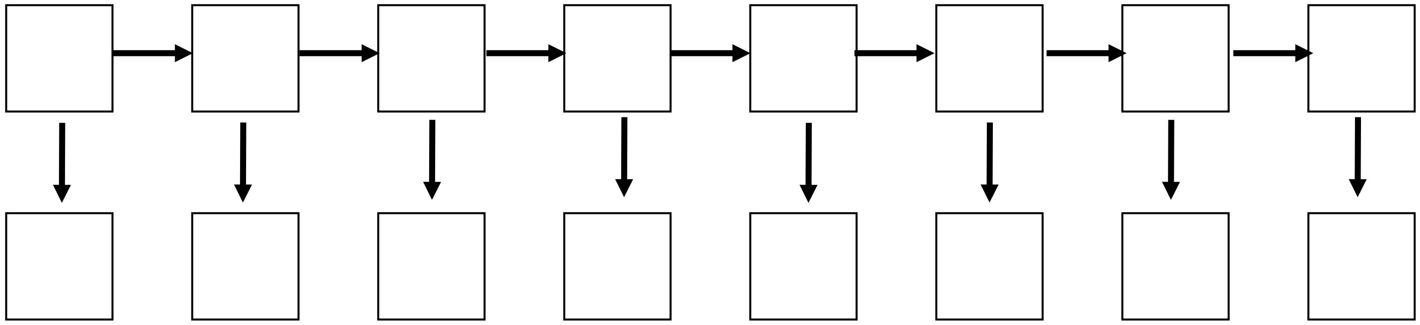
1st order Markov chain



2nd order Markov chain



1st order with stochastic observations -- HMM



Order & Conditional Probabilities

Order

0th **P(ACTGTC) = p(A) x p(C) x p(T) x p(G) x p(T) ...**

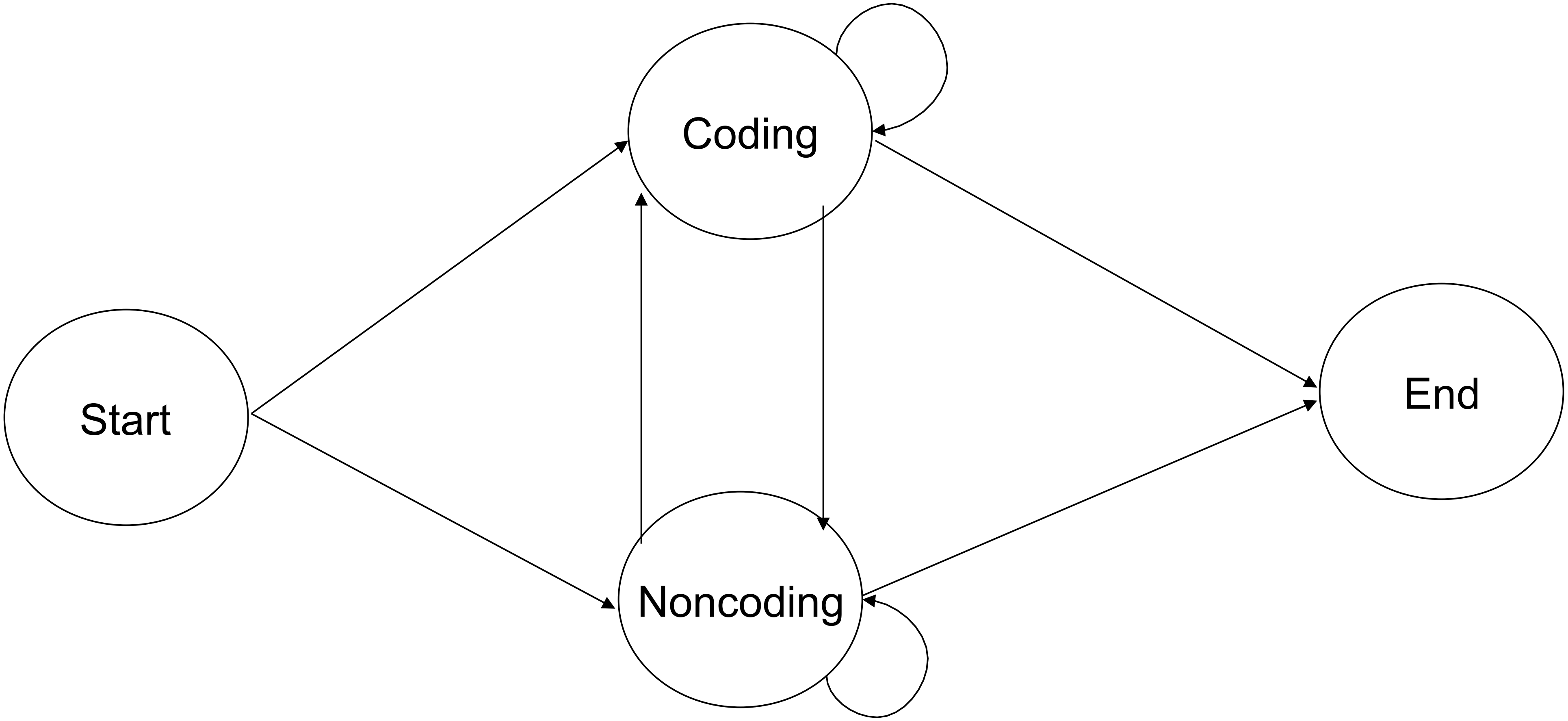
1st **P(ACTGTC) = p(A) x p(C|A) x p(T|C) x p(G|T) …**

2nd **P(ACTGCG) = p(A) x p(C|A) x p(T|AC) x p(G|CT)...**

P(T|AC)

Probability of T given AC

HMM - Combined Model for Gene Detection



1st-order transition matrix (4x4)

	A	C	G	T
A	0.2	0.15	0.25	0.2
C	0.3	0.35	0.25	0.2
G	0.3	0.4	0.3	0.3
T	0.2	0.1	0.2	0.2

2nd Order Model (16x4)

	A	C	G	T
AA	0.1	0.3	0.25	0.05
AC	0.05	0.25	0.3	0.1
AG	0.3	0.05	0.1	0.25
AT	0.25	0.1	0.05	0.3

.

.

.

Three Basic Problems of HMMs

- ✓ The Evaluation Problem
  - ✓ Given an HMM and a sequence of observations, what is the probability that the observations are generated by the model?
- ✓ The Decoding Problem
  - ✓ Given a model and a sequence of observations, what is the most likely state sequence in the model that produced the observations?
- ✓ The Learning Problem
  - ✓ Given a model and a sequence of observations, how should we adjust the model parameters in order to maximize

What Questions can an HMM Answer?

Viterbi Algorithm:

What is the most probable path that generated sequence X?

Forward Algorithm:

What is the likelihood of sequence X given HMM M – $\Pr(\mathrm{X}|\mathrm{M})$?

Forward-Backward (Baum-Welch) Algorithm:

What is the probability of a particular state k having generated symbol X_i?

“Decoding” With HMM

Given observations, we would like to predict a sequence of hidden states that is most likely to have generated that sequence

Pathogenicity Island Example

Given a nucleotide sequence, we want a labeling of each nucleotide as either “pathogenicity island” or “background DNA”

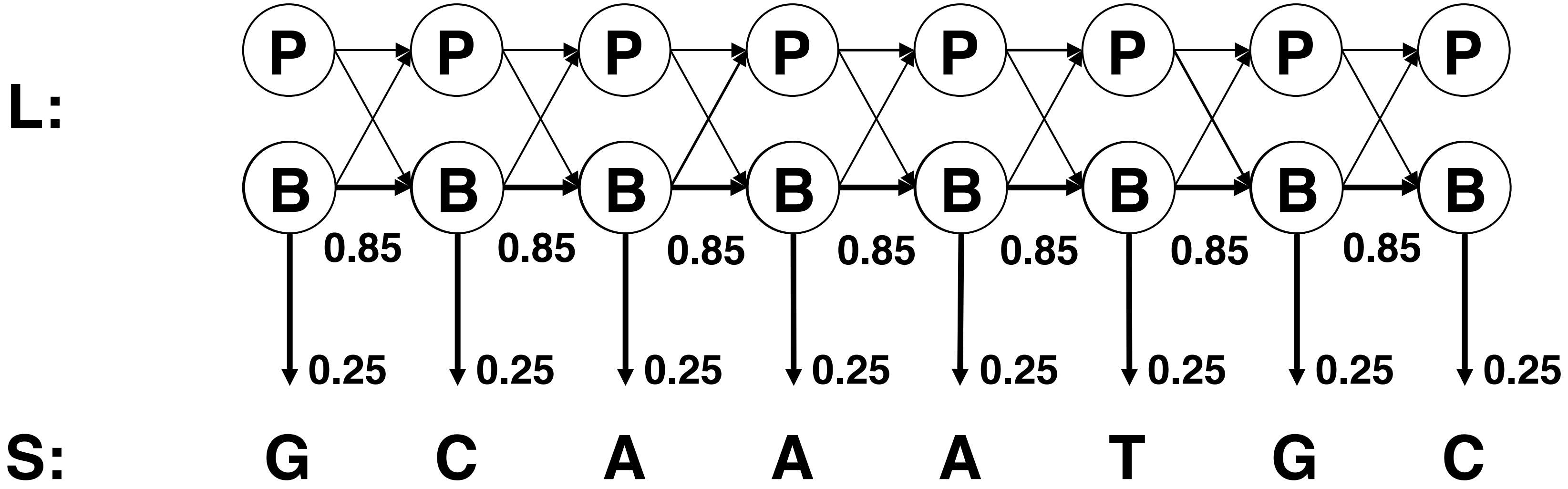
The Most Likely Path

✓ Given observations, one reasonable choice for labeling the hidden states is:

$$L^* = \arg\max_{labels} P(Labels, Sequence \mid Model)$$

The sequence of hidden state labels, L*, (or path) that makes the labels and sequence most likely given the model

Probability of a Path,Seq

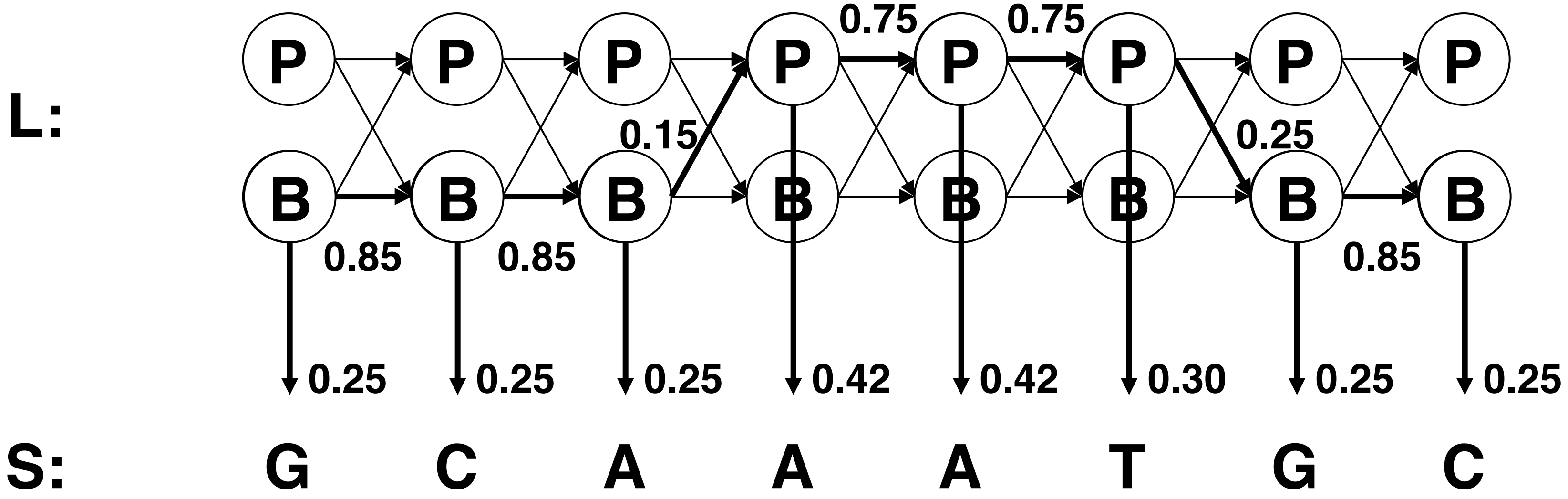


$$P = P(G \mid B)P(B_1 \mid B_0)P(C \mid B)P(B_2 \mid B_1)P(A \mid B)P(B_3 \mid B_2)\ldots P(C \mid B_7)$$

$$= (0.85)^7 \times (0.25)^8$$

$$= 4.9 \times 10^{-6}$$

Probability of a Path,Seq



$$
\begin{aligned}
P &= P(G \mid B)P(B_1 \mid B_0)P(C \mid B)P(B_2 \mid B_1)P(A \mid B)P(P_3 \mid B_2)\ldots P(C \mid B_7) \\
&= (0.85)^3 \times (0.25)^6 \times (0.75)^2 \times (0.42)^2 \times 0.30 \times 0.15 \\
&= 6.7 \times 10^{-7}
\end{aligned}
$$

We could try to calculate the probability of every path, but….

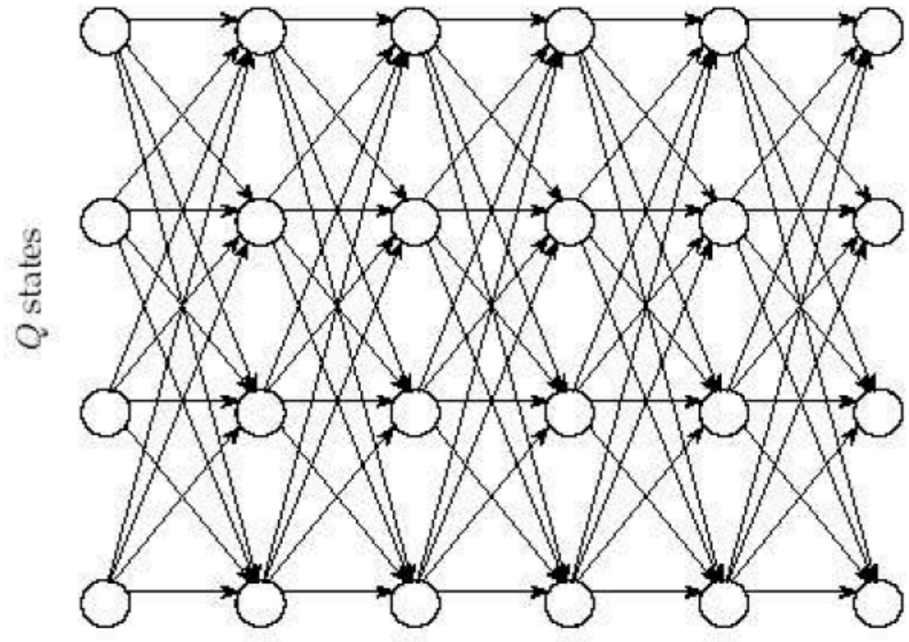
Decoding

- ✓Viterbi Algorithm
  - ✓Finds most likely sequence of hidden states or labels, L* or P* or π*, given sequence and model

$$L^{*} = \underset{labels}{\arg\max}\, P(Labels, Sequence \mid Model)$$

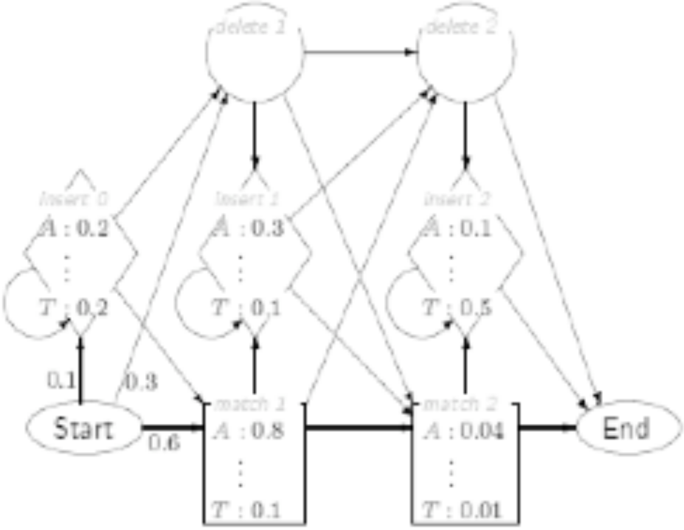
  - ✓Uses *dynamic programming* (same technique used in sequence alignment)
  - ✓Much more efficient than searching every path

Finding Best Path

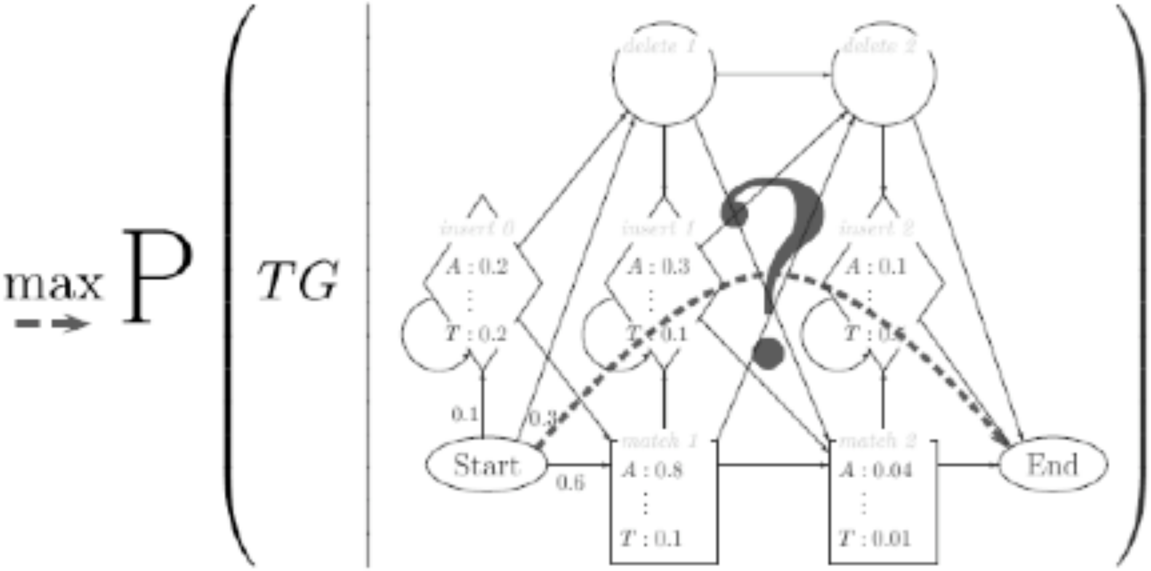


- ✓Viterbi
- ✓Dynamic programming
- ✓Maximize Probability Emission of observations on trace-back

Viterbi Algorithm

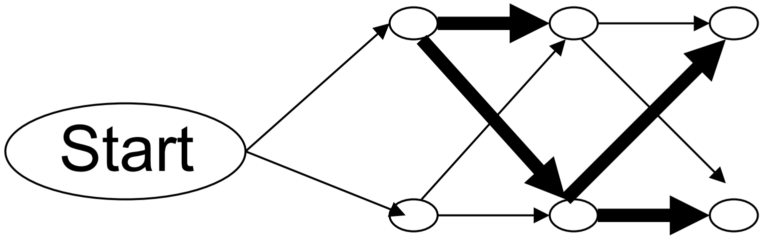


Most probable state path given sequence (observations)?

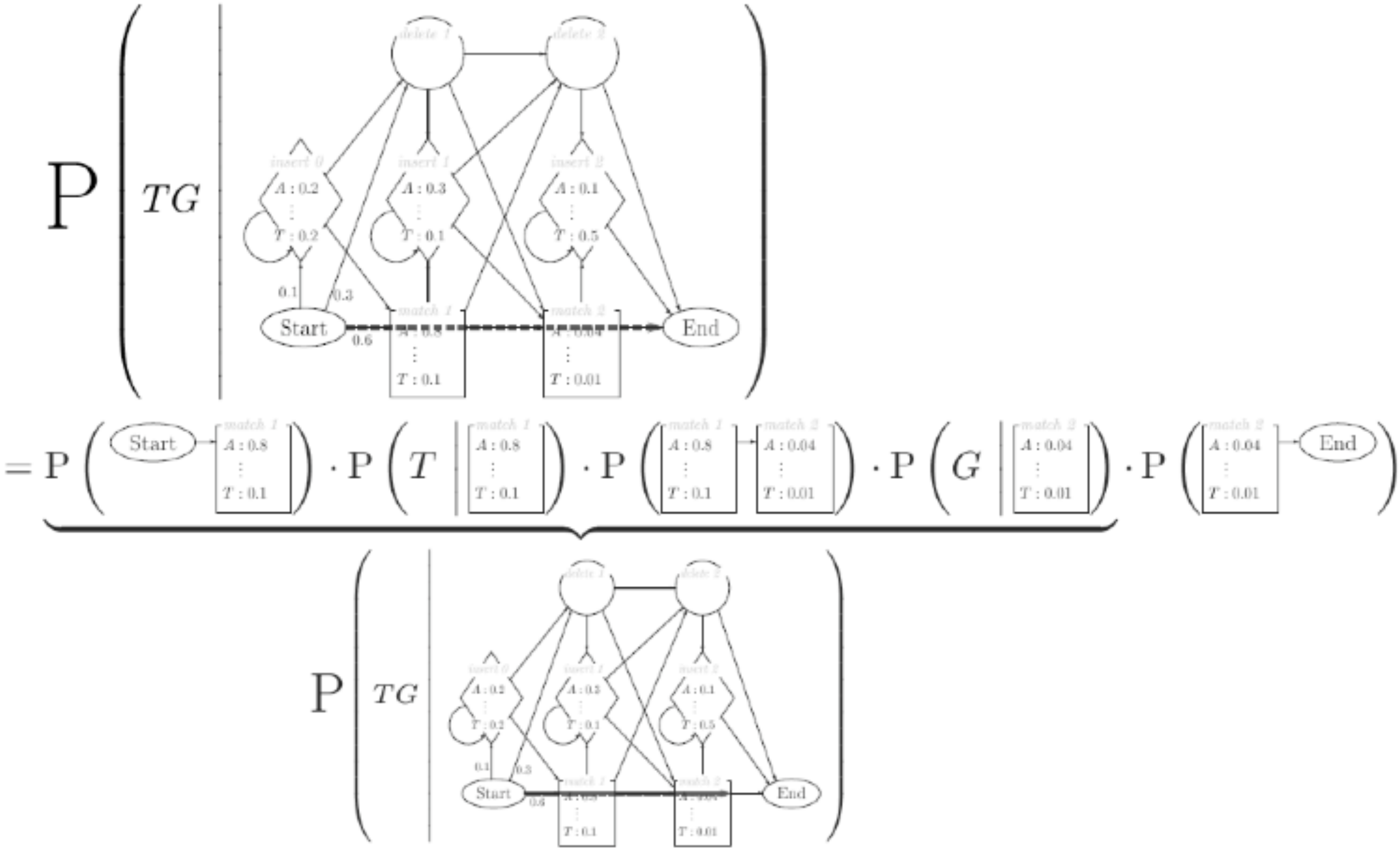


Viterbi (in pseudocode)

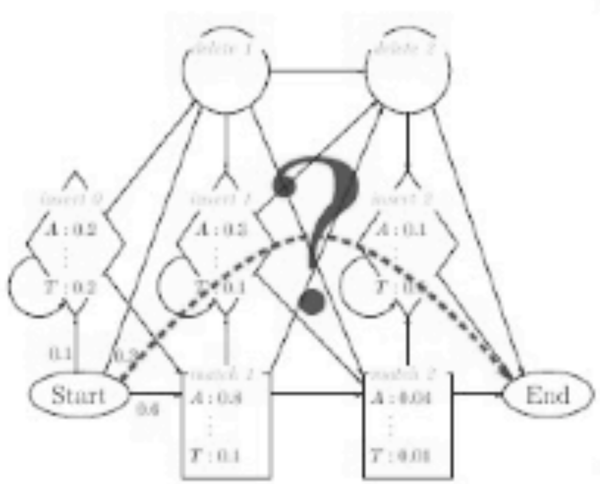
- ✓ l is previous state and k is next state
- ✓ $v_l(i)= e_l(x_i) \max_k(v_k(i-1)a_{kl})$
- ✓ π^* are the paths that maximizes the probability of the previous path times new transition in $\max_k(v_k(i-1)a_{kl})$



Each node picks one max


P
TG
delete 1
delete 2
insert 0
A : 0.2
T : 0.2
insert 1
A : 0.3
T : 0.1
insert 2
A : 0.1
T : 0.5
0.1
0.3
0.6
Start
match 1
A : 0.8
T : 0.1
match 2
A : 0.04
T : 0.01
End
= P
· P
T
G

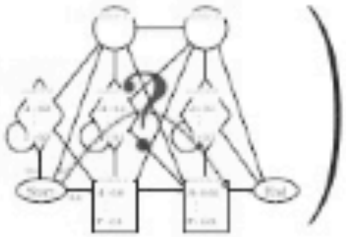
$\max\limits_{\dashrightarrow} \mathrm{P}\Big(TG \;\Big|$



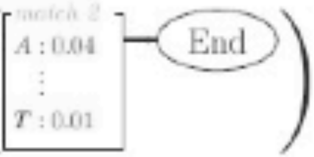
$\Big)$

$= \max \Bigg\{$

$\max\limits_{\dashrightarrow} \mathrm{P}\Big(TG \;\Big|$

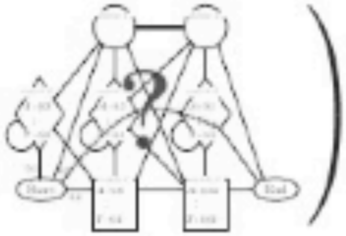


$\Big) \cdot \mathrm{P}\Big($

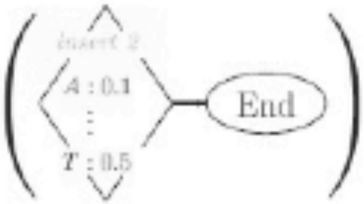


$\Big)$

$\max\limits_{\dashrightarrow} \mathrm{P}\Big(TG \;\Big|$

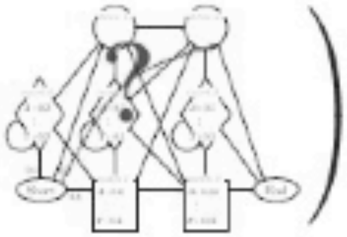


$\Big) \cdot \mathrm{P}\Big($

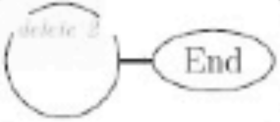


$\Big)$

$\max\limits_{\dashrightarrow} \mathrm{P}\Big(TG \;\Big|$

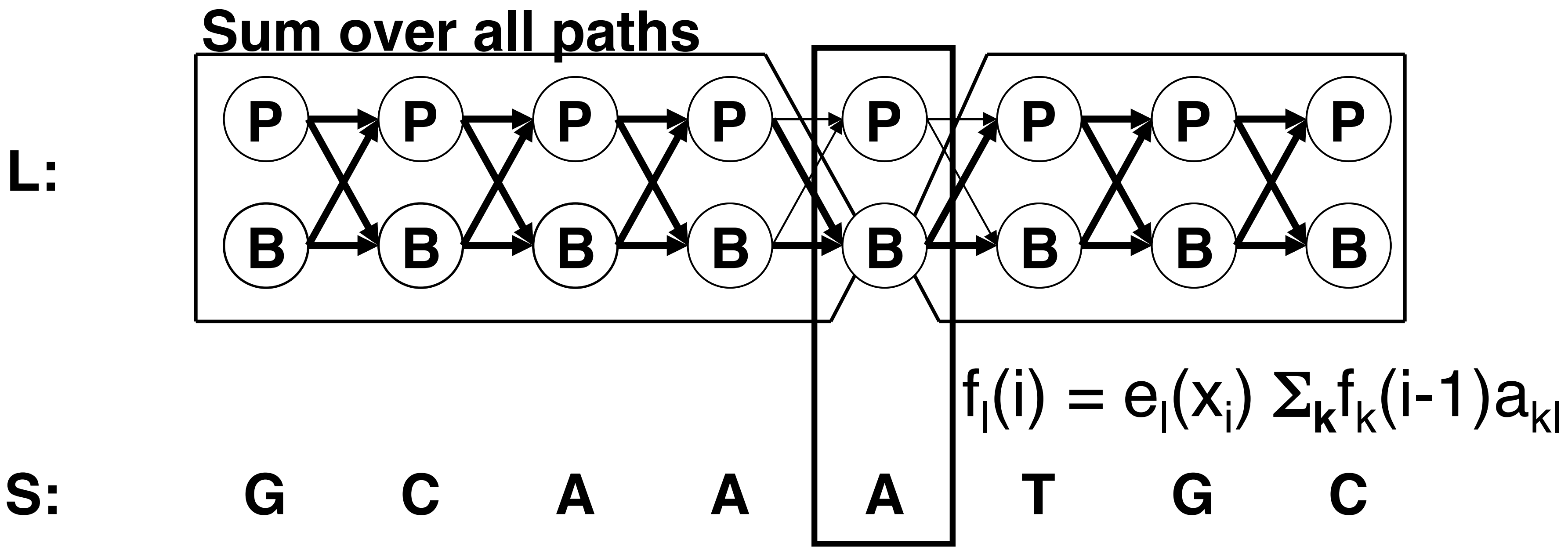


$\Big) \cdot \mathrm{P}\Big($



$\Big)$

Forward Alg: Probability of a Single Label (Hidden State)



$$f_l(i) = e_l(x_i)\ \Sigma_{\mathbf{k}} f_k(i-1)a_{kl}$$

Forward algorithm

(dynamic programming)

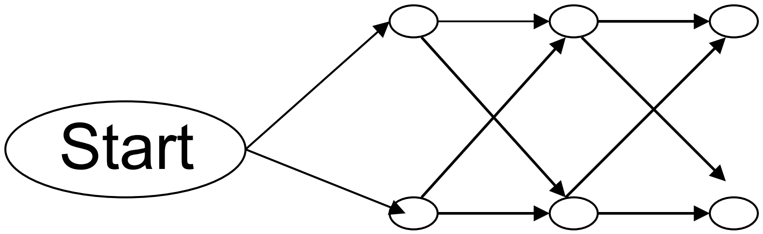
P(Label$_5$=B|S)

✓ Calculate most probable label, L^*_i , at each position i

✓ Do this for all N positions gives us $\{L^*_1, L^*_2, L^*_3 \ldots L^*_N\}$

Forward Algorithm

$$f_l(i) = e_l(x_i)\, \boldsymbol{\Sigma}_{\mathbf{k}} f_k(i\text{-}1)a_{kl}$$

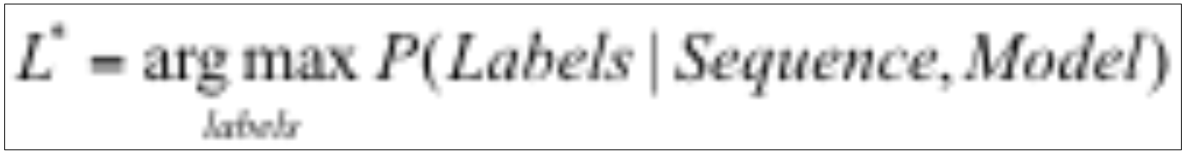


$$P(x) = \boldsymbol{\Sigma}_{\mathbf{k}}\ f_k(N)a_{k0}$$

Add probs of all
Different paths to get
Probability of sequence

Two Decoding Options

- ✓Viterbi Algorithm
  - ✓Finds most likely sequence of hidden states, L* or P* or π*, given sequence and model



$$L^* = \underset{labels}{\arg\max}\, P(Labels \mid Sequence, Model)$$

- ✓Posterior Decoding
  - ✓Finds most likely label at each position for all positions, given sequence and model

$$\{L^*_1, L^*_2, L^*_3 \ldots L^*_N\}$$

  - ✓Forward and Backward equations

Relation between Viterbi and Forward

VITERBI

$V_j(i)$ = P(most probable path ending in state *j* with observation *i*)

Initialization:

$V_0(0) = 1$

$V_k(0) = 0$, for all $k > 0$

Iteration:

$V_l(i)= e_l(x_i)\mathbf{max_k} V_k(i-1)\ a_{kl}$

Termination:

$P(x, \pi^*) = \mathbf{max_k} V_k(N)$

FORWARD

$f_l(i)=P(x_1\ldots x_i,state_i=l)$

Initialization:

$f_0(0) = 1$

$f_k(0) = 0$, for all $k > 0$

Iteration:

$f_l(i) = e_l(x_i)\ \boldsymbol{\Sigma_k} f_k(i-1)a_{kl}$

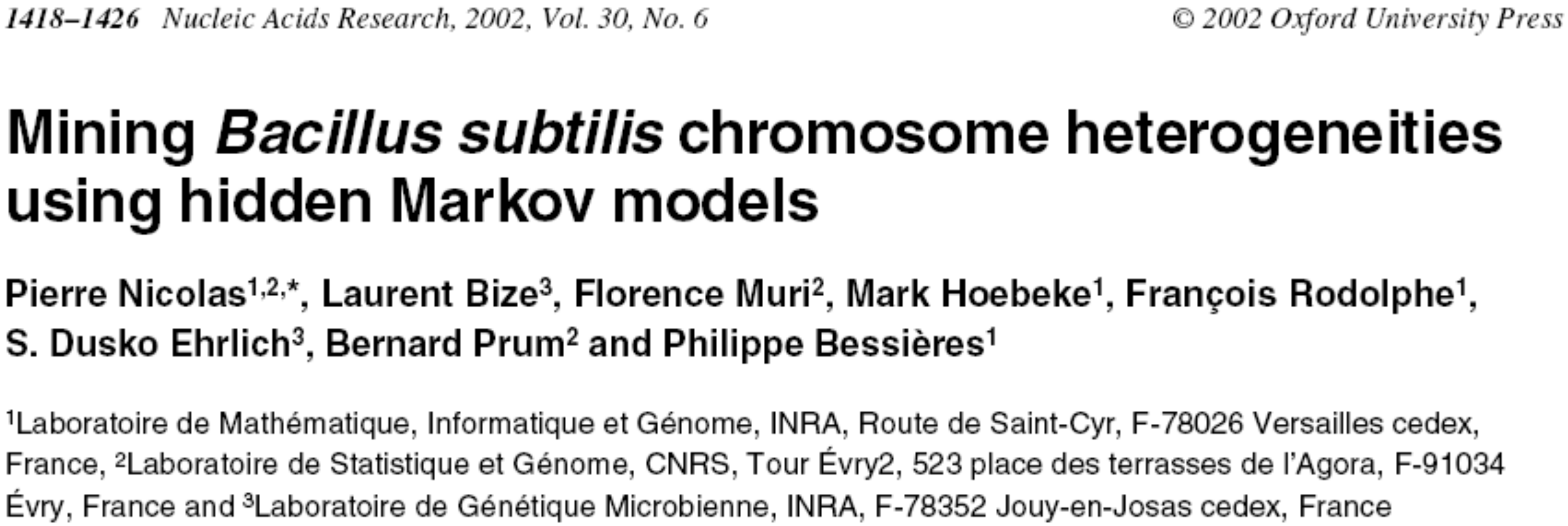
Termination:

$P(x) = \boldsymbol{\Sigma_k}\ f_k(N)a_{k0}$

Forward/Backward Algorithms

- ✓ Way to compute probability of most probable path
- ✓ Forward and Backward can be combined to find Probability of emission, x_i from state k given sequence x. $P(\pi_i=k \mid x)$
- ✓ $P(\pi_i=k \mid x)$ is called posterior decoding
- ✓ $P(\pi_i=k \mid x) = f_k(l)b_k(l)/P(x)$

Example Application: *Bacillus subtilis*



1418–1426 *Nucleic Acids Research, 2002, Vol. 30, No. 6* *© 2002 Oxford University Press*

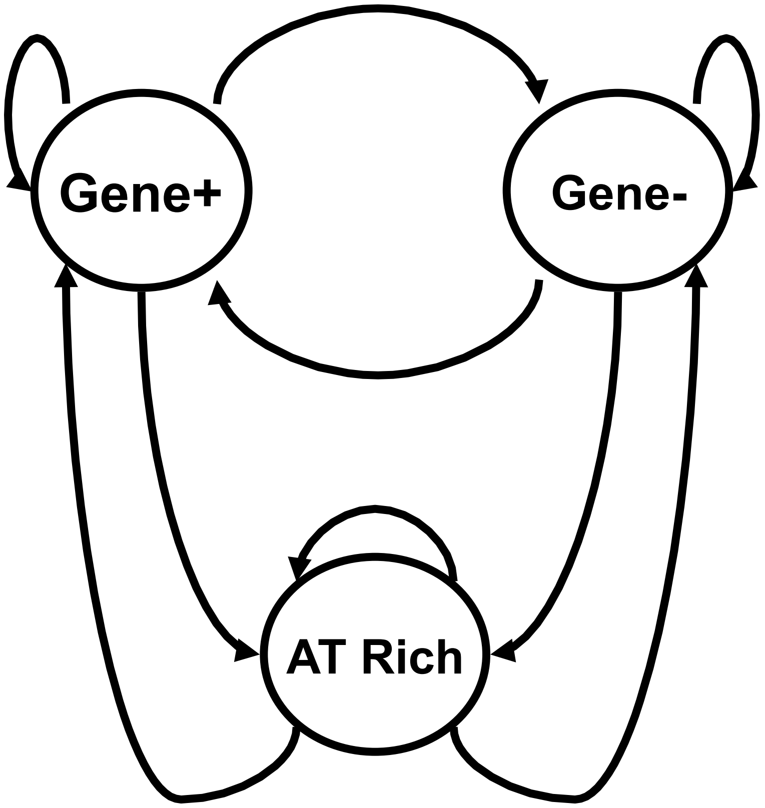
Mining *Bacillus subtilis* chromosome heterogeneities using hidden Markov models

Pierre Nicolas[1,2,*], Laurent Bize[3], Florence Muri[2], Mark Hoebeke[1], François Rodolphe[1], S. Dusko Ehrlich[3], Bernard Prum[2] and Philippe Bessières[1]

[1]Laboratoire de Mathématique, Informatique et Génome, INRA, Route de Saint-Cyr, F-78026 Versailles cedex, France, [2]Laboratoire de Statistique et Génome, CNRS, Tour Évry2, 523 place des terrasses de l'Agora, F-91034 Évry, France and [3]Laboratoire de Génétique Microbienne, INRA, F-78352 Jouy-en-Josas cedex, France

Method

Three State Model



Second Order Emissions

$P(S_i)=P(S_i|\text{State},S_{i-1},S_{i-2})$
(capturing trinucleotide Frequencies)

Train using EM

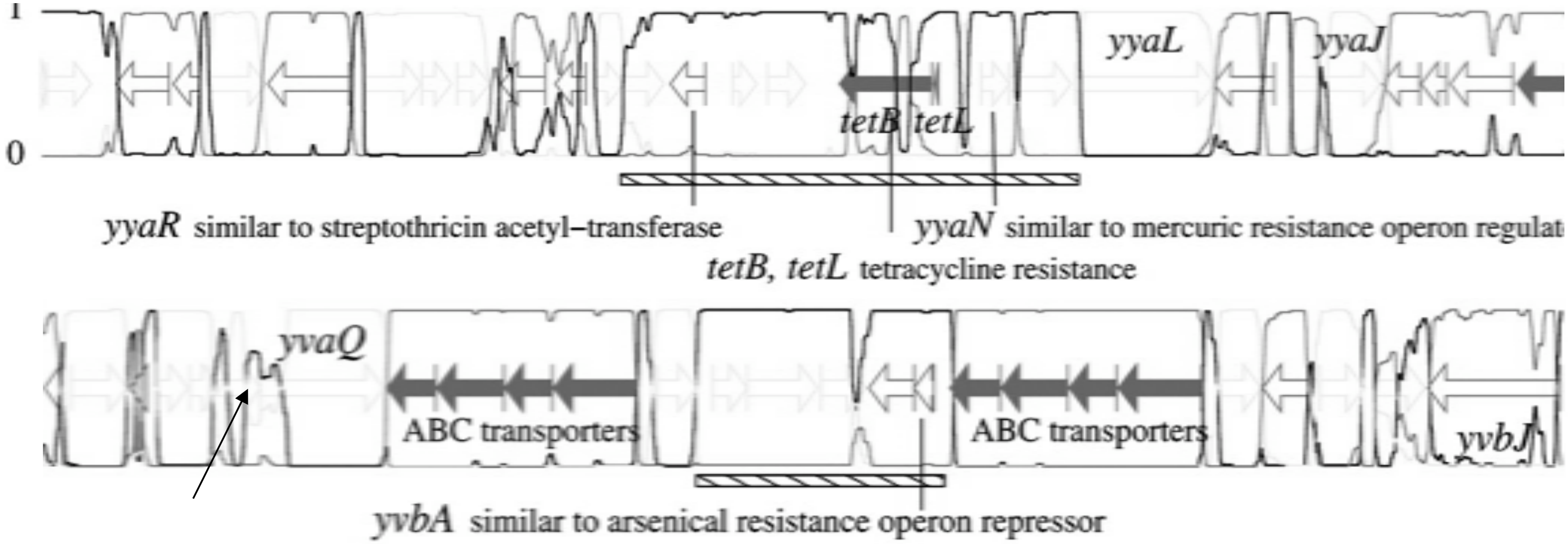
Predict w/Posterior Decoding

Nicolas et al (2002) NAR

Results

Gene on positive strand

Gene on negative strand



A/T Rich
- **Intergenic regions**
- **Islands**

Each line is P(label|S,model) color coded by label

Nicolas et al (2002) NAR

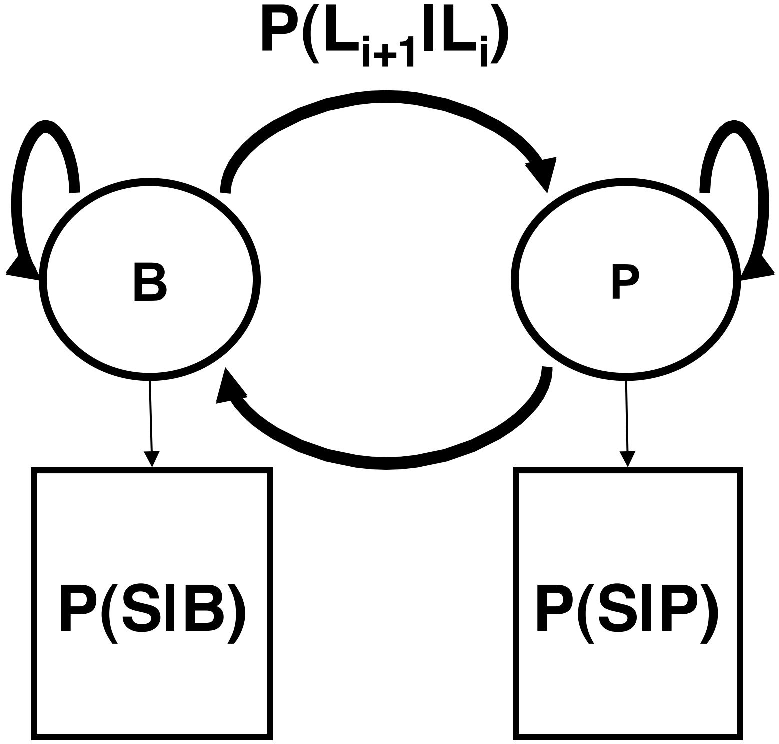
Training an HMM

Transition probabilities

e.g. $P(P_{i+1}|B_i)$ – the probability of entering a pathogenicity island from background DNA

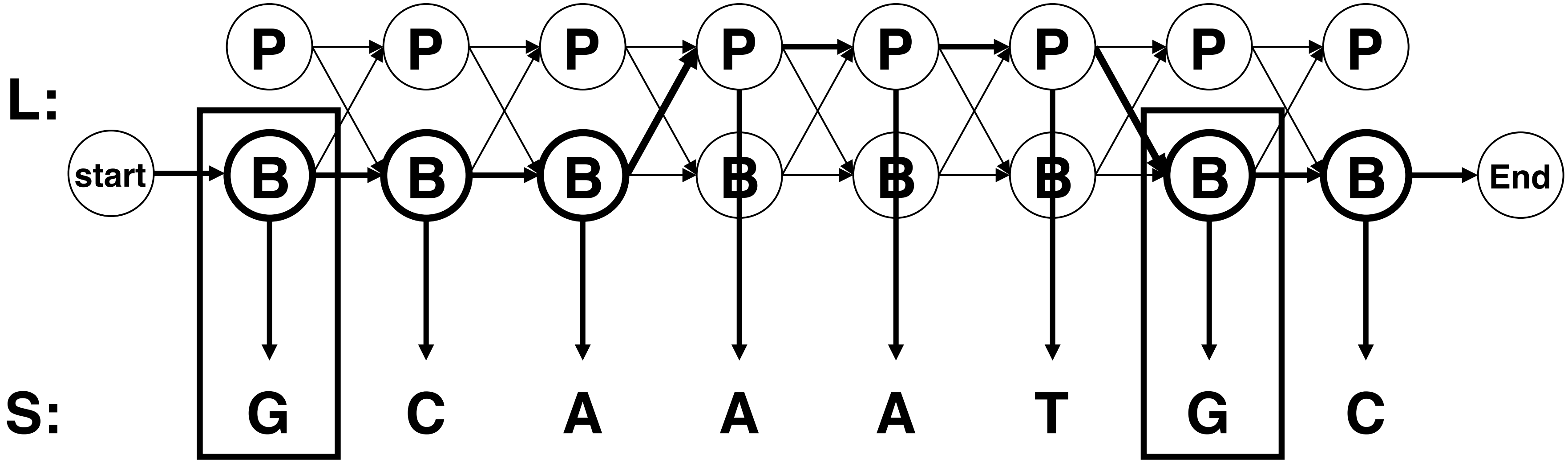
Emission probabilities

i.e. the nucleotide frequencies for background DNA and pathogenicity islands



Learning From Labelled Data

If we have a sequence that has islands marked, we can simply count



$P(L_{i+1}|L_i)$

	B_{i+1}	P_{i+1}	End
B_i	3/5	1/5	1/5
P_i	1/3	2/3	0
Start	1	0	0

P(S|B)

A:	1/5
T:	0
G:	2/5
C:	2/5

!

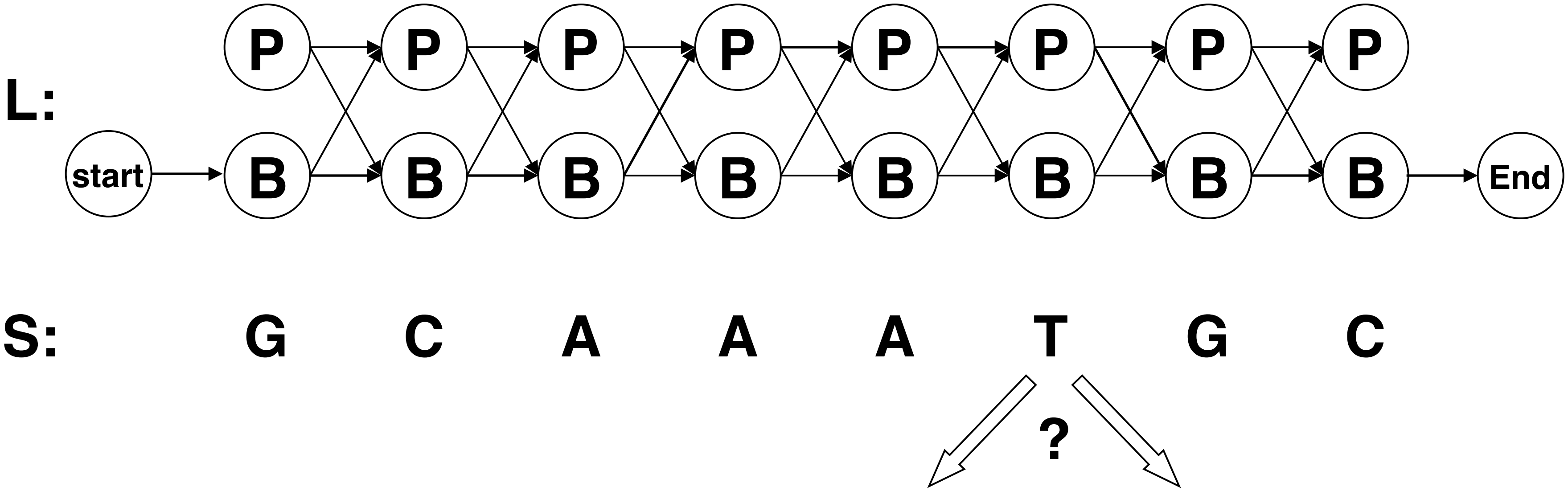
P(S|P)

A:
T:
G:
C:

ETC..

Unlabelled Data

How do we know how to count?



$P(L_{i+1}|L_i)$

	B_{i+1}	P_{i+1}	End
B_i			
P_i		?	
Start			

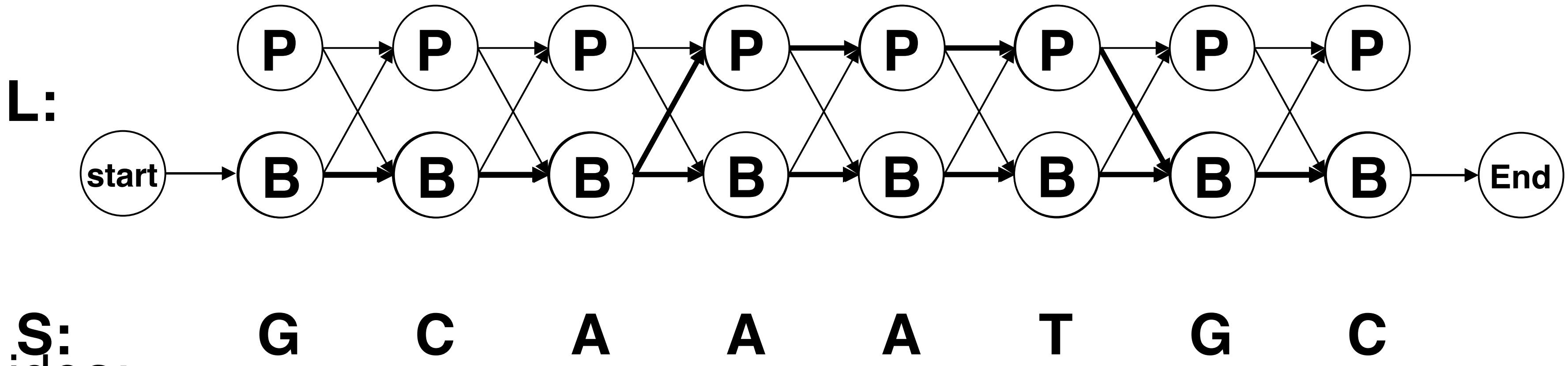
$P(S|B)$

A:
T:
G:
C:

$P(S|P)$

A:
T:
G:
C:

Unlabeled Data



An idea:

1. Imagine we start with some parameters

   (e.g. initial or bad model)

2. We *could* calculate the most likely path, P*, given those parameters and S

3. We *could* then use P* to recalculate our parameters by maximum likelihood

4. And iterate (to convergence)

~~$P(L_{i+1}|L_i)^0 P(S|B)^0 P(S|P)^0$~~

~~$P(L_{i+1}|L_i)^1 P(S|B)^1 P(S|P)^1$~~

$P(L_{i+1}|L_i)^2 P(S|B)^2 P(S|P)^2$

…

$P(L_{i+1}|L_i)^K P(S|B)^K P(S|P)^K$

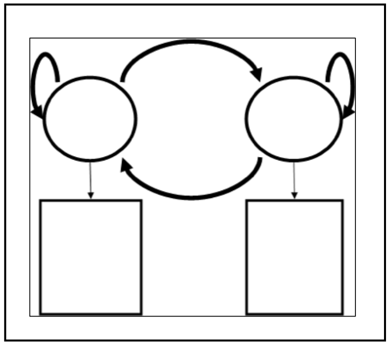
Training Models for Classification

- ✓ Correct Order for the model
- ✓ Higher order models remember more “history”
- ✓ Additional history can have predictive value
  - ✓ Example:
  - ✓ predict the next word in this sentence fragment
  - ✓ “…finish __” (up, it, first, last, …?)
  - ✓ now predict it given more history
  - ✓ “Fast guys finish __”

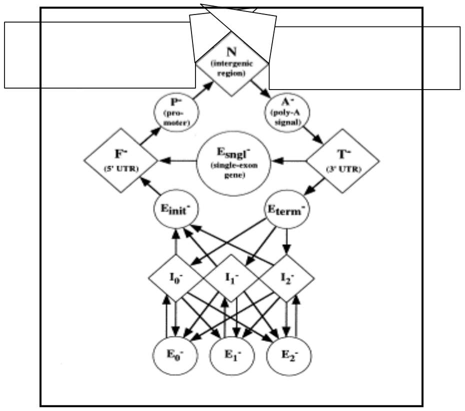
Model Order

- ✓ However, the number of parameters to estimate grows exponentially with the order for modeling DNA we need parameters for an nth order model, with n>=5 normally
- ✓ The higher the order, the less reliable we can expect our parameter estimates to be
  - ✓ estimating the parameters of a 2nd order Markov chain from the complete genome of E. Coli, each word > 72,000 times on average
  - ✓ estimating the parameters of an 8th order chain, word 5 times on average

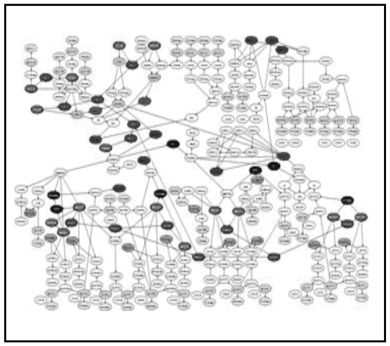
HMMs in Context



- ✓ HMMs
  - ✓ Sequence alignment
  - ✓ Gene Prediction

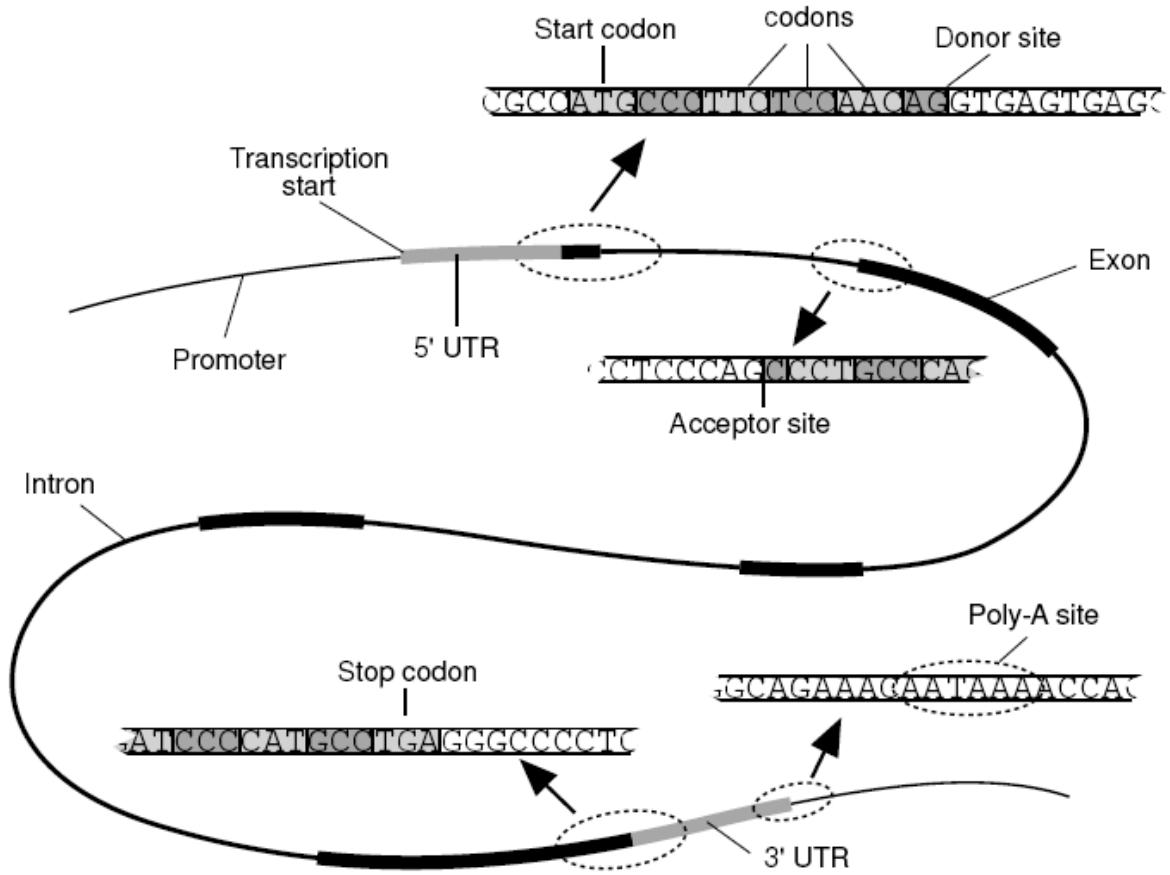


- ✓ Generalized HMMs
  - ✓ Variable length states
  - ✓ Complex emissions models
  - ✓ *e.g. Genscan*



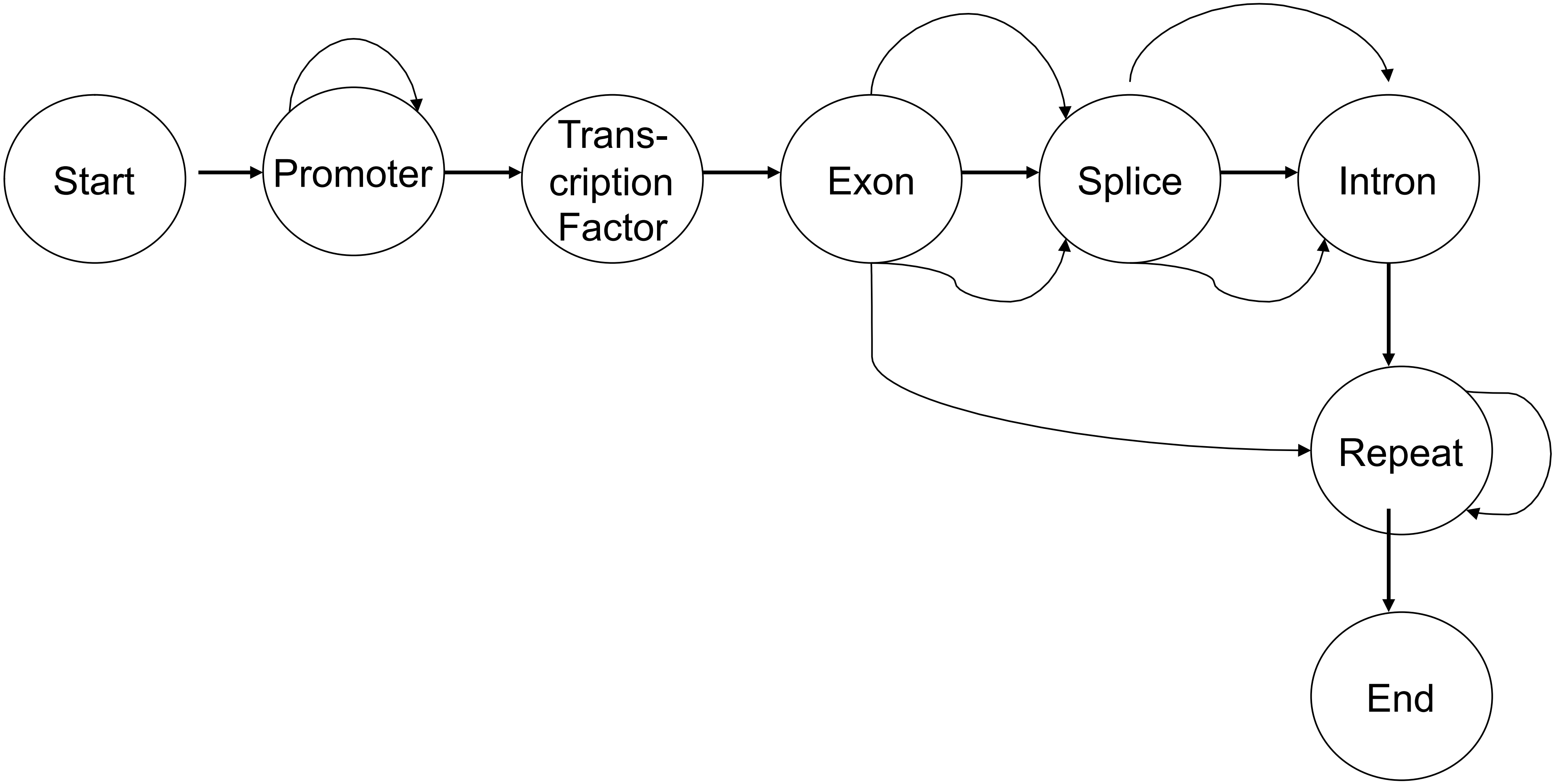
- ✓ Bayesian Networks
  - ✓ General graphical model
  - ✓ Arbitrary graph structure
  - ✓ e.g. *Regulatory network analysis*

HMMs can model different regions

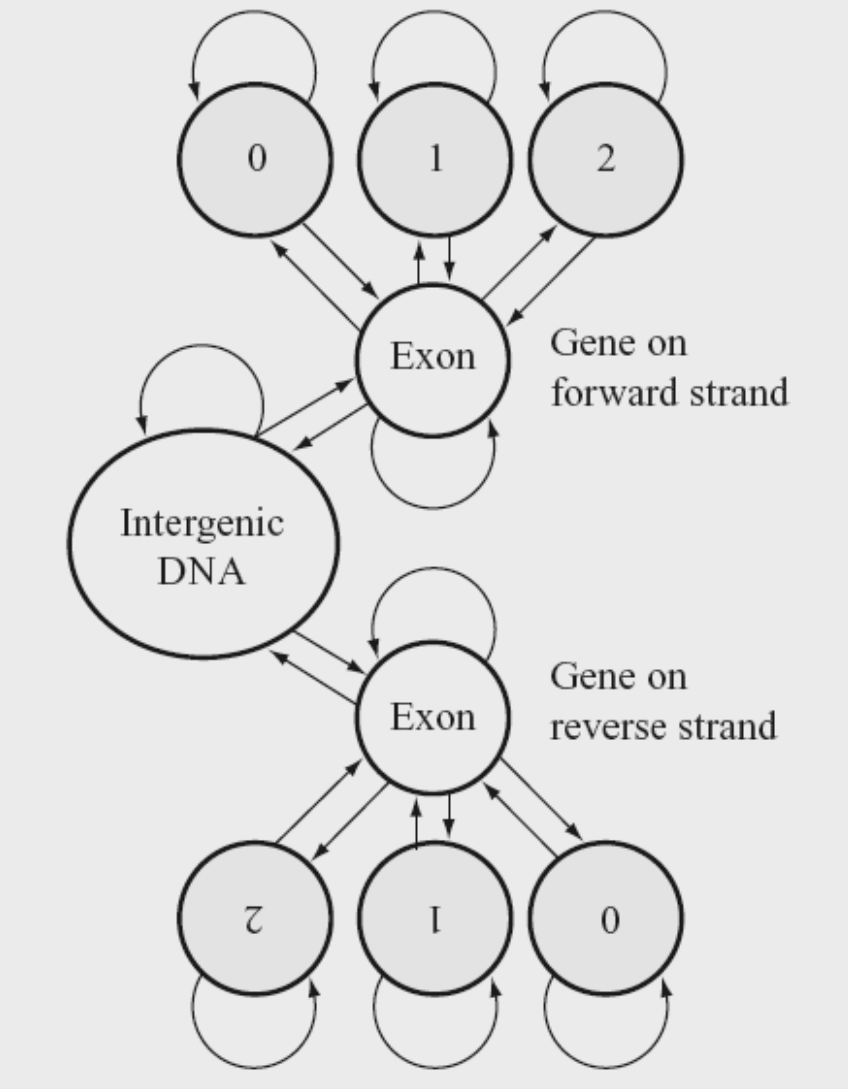


igure 4.8: The structure of a gene with some of the important signals shown.

Example Model for Gene Recognition



Another Example



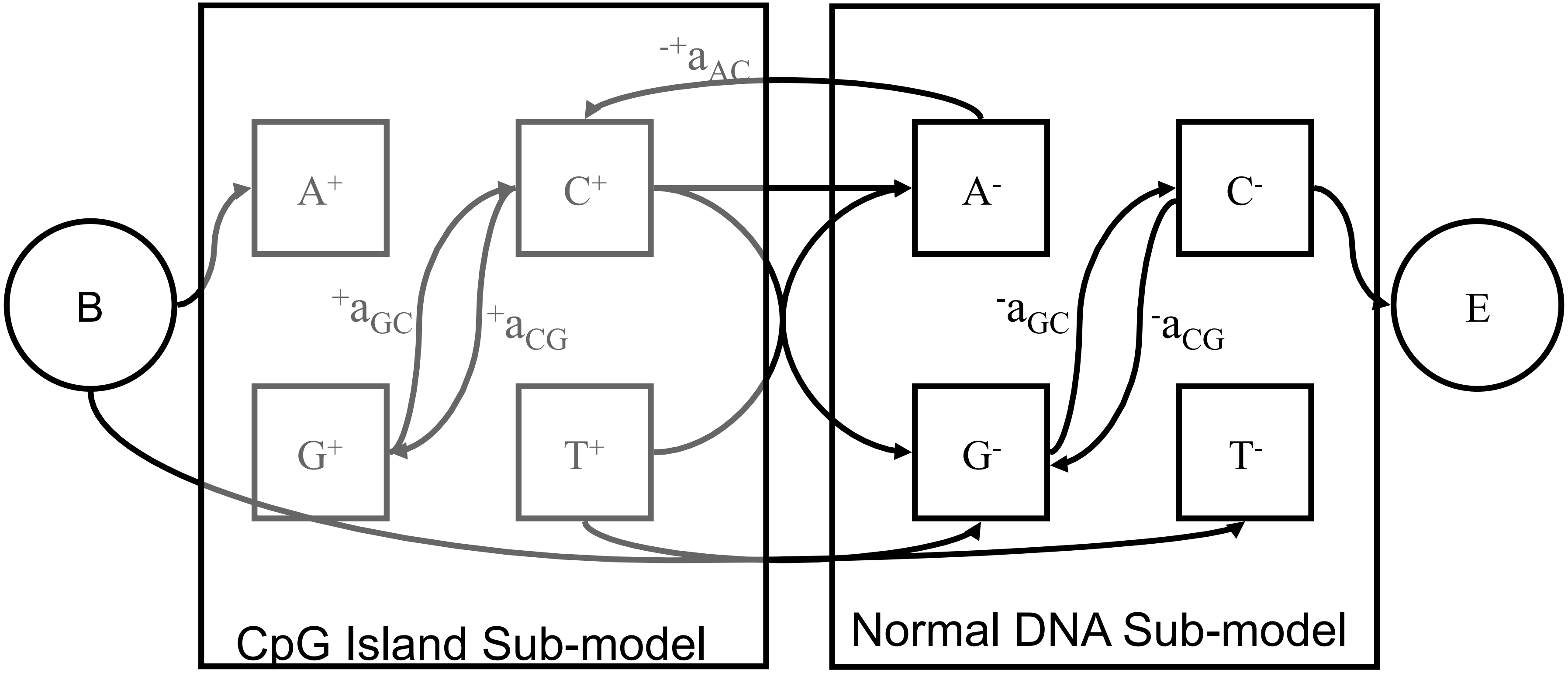
CpG Islands: Another Application

- ✓ CG dinucleotides are rarer in eukaryotic genomes than expected given the independent probabilities of C, G
- ✓ Particularly, the regions upstream of genes are richer in CG dinucleotides than elsewhere - *CpG islands*

CpG Islands

CpG island DNA states:
large C, G transition probabilities

"Normal DNA" states:
small C, G transition probabilities



Most transitions omitted for clarity

CpG Islands

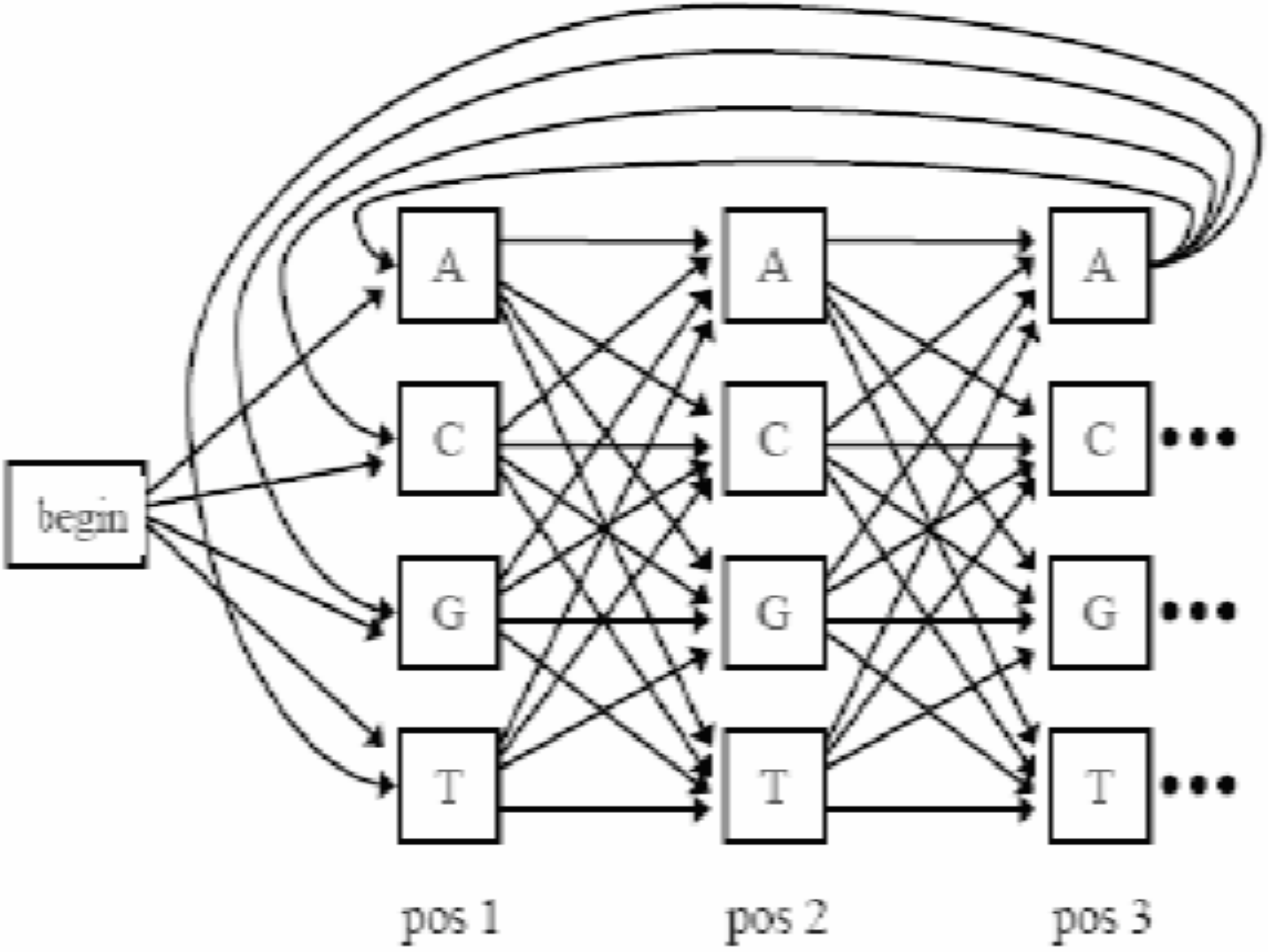
- ✓ In human genome, CG dinucleotides are relatively rare
  - ✓ CG pairs undergo a process called methylation that modifies the C nucleotide
  - ✓ A methylated C mutate (with relatively high chance) to a T
- ✓ Promotor regions are CG rich
  - ✓ These regions are not methylated, and thus mutate less often
  - ✓ These are called CG (aka CpG) islands

CpG Island Prediction

- ✓ In a CpG island, the probability of a “C” following a “G” is much higher than in “normal” intragenic DNA sequence.
- ✓ We can construct an HMM to model this by combining two HMMs: one for normal sequence and one for CpG island sequence.
- ✓ Transitions between the two *sub-models* allow the model to switch between CpG island and normal DNA.
- ✓ Because there is more than one state that can generate a given character, the states are “hidden” when you just see the sequence.
- ✓ For example, a “C” can be generated by either the $\underline{\mathrm{C}^{+}}$ or $\underline{\mathrm{C}^{-}}$ states in the following model.

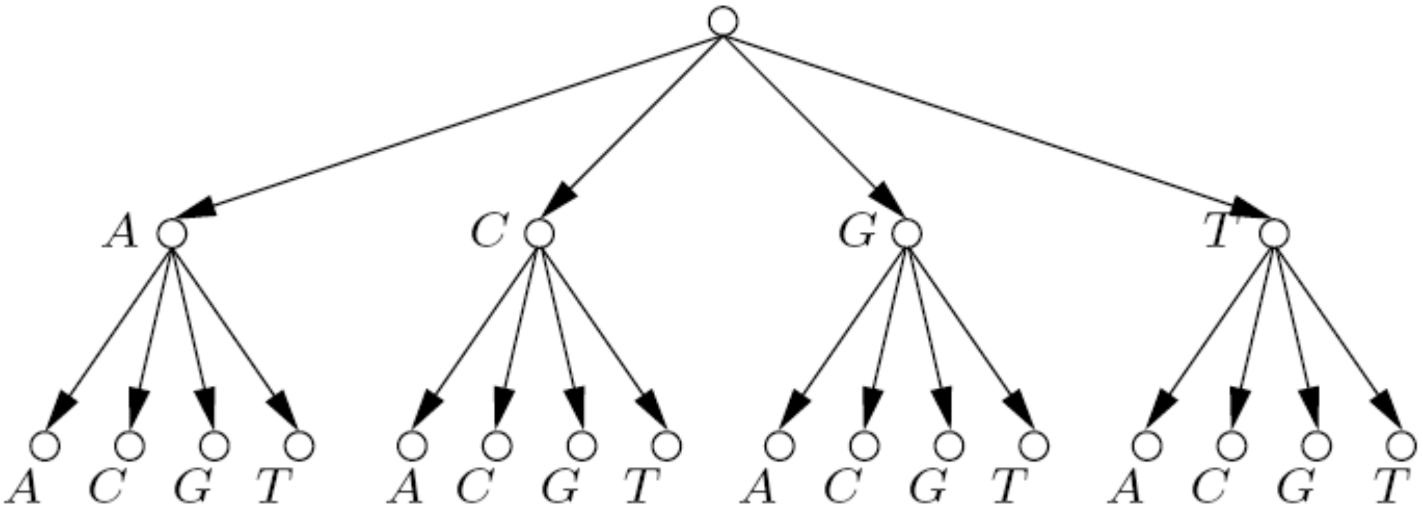
Inhomogenous Markov Chains

Borodovsky's Lab: http://exon.gatech.edu/GeneMark/

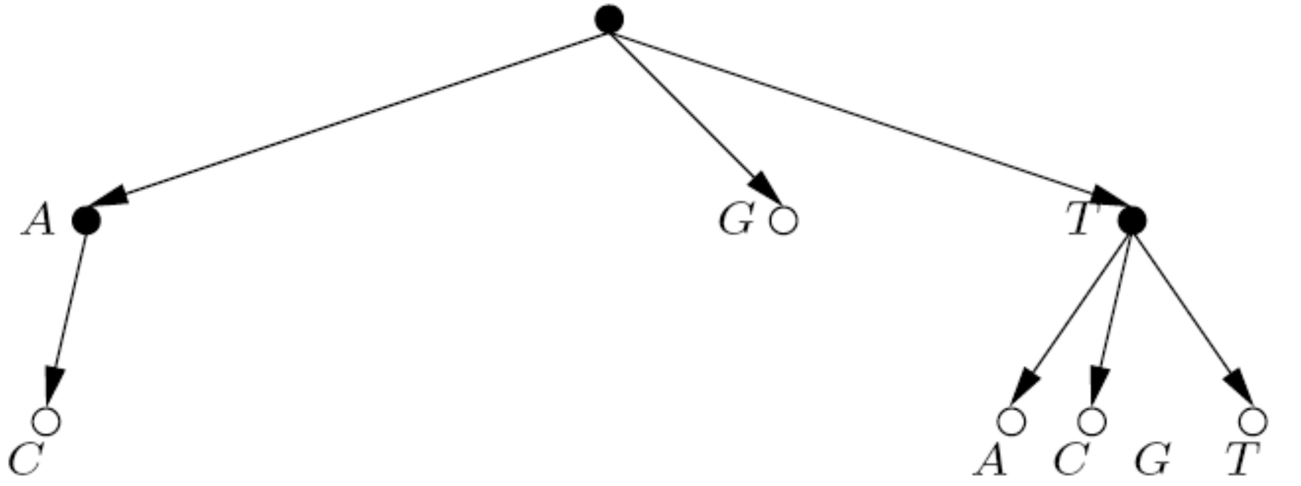


Variable-length

Full



Variable
Length



Interpolated HMMs

- ✓ Manage Model Trade-off by interpolating between various HMM Model orders
- ✓ GlimmerHMM

The Three Basic HMM Problems

✓ Problem 1 (Evaluation):

Given the observation sequence $O=o_1,\ldots,o_T$ and an HMM model, how do we compute the probability of O given the model?

✓ Problem 2 (Decoding):

Given the observation sequence $O=o_1,\ldots,o_T$ and an HMM model, how do we find the state sequence that best explains the observations?

The Three Basic HMM Problems

- ✓ Problem 3 (Learning): How do we adjust the model parameters to maximize the probability of observations given the model?

Conclusions

✓Markov Models

✓HMMs

✓Issues

✓Applications

Example of Viterbi, Forward, Backward, and Posterior Algorithms

Real DNA sequences are inhomogeneous and can be described by a hidden Markov model with hidden states representing different types of nucleotide composition. Consider an HMM that includes two hidden states H and L for high and lower C+G content, respectively. Initial probabilities for both H and L are equal to 0.5, while transition probabilities are as follows: a_{HH}=0.5, a_{HL}=0.5, a_{LL}=0.6, a_{LH}=0.4. Nucleotides T, C, A, G are emitted from states H and L with probabilities 0.2, 0.3, 0.2, 0.3, and 0.3, 0.2, 0.3, 0.2, respectively. Use the Viterbi algorithm to define the most likely sequence of hidden states for the sequence, X=TGC.