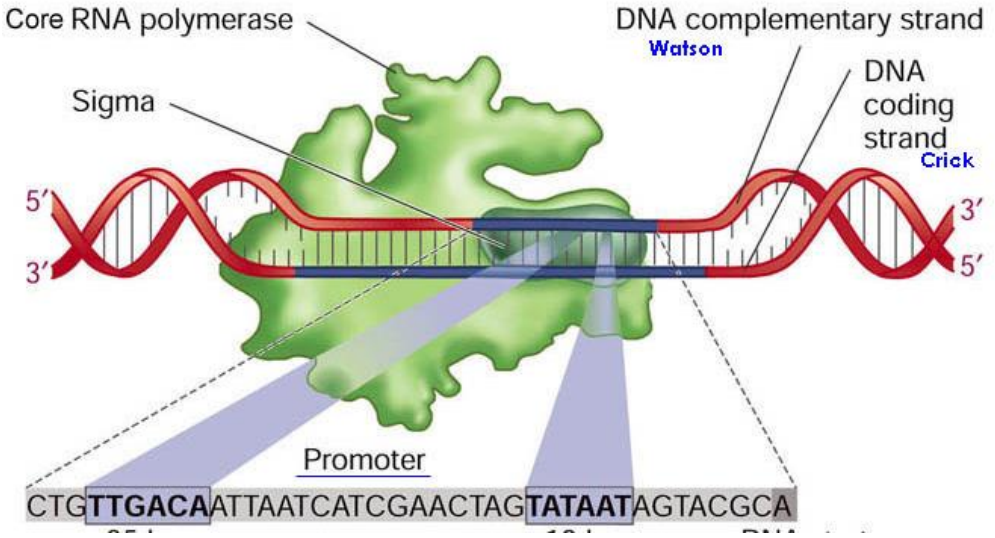
Markov Models

BMI/CS 576
www.biostat.wisc.edu/bmi576/
Mark Craven
craven@biostat.wisc.edu

Motivation for sequence modeling



these sequences are E. coli promoters

```
tctgaaatgagctgttgacaattaatcatcgaactagttaactagtacgcaagttca
accggaagaaaaccgtgacattttaacacgtttgttacaaggtaaaggcgacgccgc
aaattaaaatttttattgacttaggtcactaaatactttaaccaatataggcatagcg
ttgtcataatcgacttgtaaaccaaattgaaaagatttaggtttacaagtctacacc
catcctcgcaccagtcgacgacggtttacgctttacgtatagtggcgacaatttttt
tccagtataatttgttggcataattaagtacgacgagtaaaattacatacctgcccg
acagttatccactattcctgtggataaccatgtgtattagagttagaaaacacgagg
```

these sequences are not promoters

```
atagtctcagagtcttgacctactacgccagcattttggcggtgtaagctaaccatt
aactcaaggctgatacggcgagacttgcgagccttgtccttgcggtacacagcagcg
ttactgtgaacattattcgtctccgcgactacgatgagatgcctgagtgcttccgtt
tattctcaacaagattaaccgacagattcaatctcgtggatggacgttcaacattga
aacgagtcaatcagaccgctttgactctggtattactgtgaacattattcgtctccg
aagtgcttagcttcaaggtcacggatacgaccgaagcgagcctcgtcctcaatggcc
gaagaccacgcctcgccaccgagtagacccttagagagcatgtcagcctcgacaact
```

How can we tell the difference? Is this sequence a promoter?

```
ccatcaaaaaaatattctcaacataaaaaactttgtgtaatacttgtaacgctacat
```

Motivation for Markov models in computational biology

- there are many cases in which we would like to represent the statistical regularities of some class of sequences
  - genes
  - various regulatory sites in DNA (e.g. promoters)
  - proteins in a given family
  - etc.

- Markov models are well suited to this type of task

Markov chain models

- a Markov chain model is defined by
  - a set of states
    - some states *emit* symbols
    - other states (e.g. the *begin* and *end*) are *silent*
  - a set of transitions with associated probabilities
    - the transitions emanating from a given state define a distribution over the possible next states

Markov chain models

- Let X be a sequence of random variables X_1 ... X_L representing a biological sequence
- from the chain rule of probability

$$
\begin{aligned}
P(X) &= P(X_L, X_{L-1}, ..., X_1) = \\
&= P(X_L \mid X_{L-1}, ..., X_1) \times \\
&\quad P(X_{L-1} \mid X_{L-2}, ..., X_1) \times \\
&\quad \vdots \\
&\quad P(X_1)
\end{aligned}
$$

Markov chain models

- from the chain rule we have

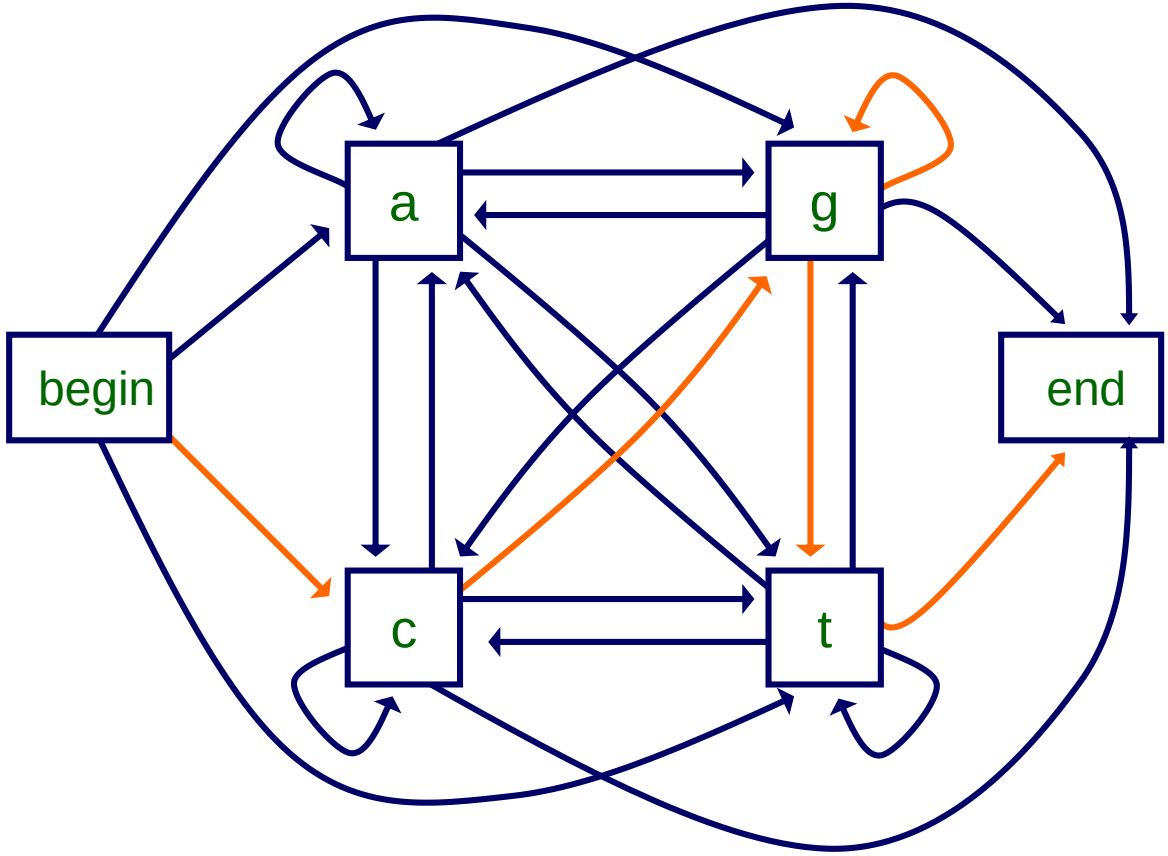
$$P(X) = P(X_L \mid X_{L-1}, \ldots, X_1)\, P(X_{L-1} \mid X_{L-2}, \ldots, X_1) \text{ K } P(X_1)$$

- key property of a (1st order) Markov chain: the probability of each X_i depends only on the value of X_{i-1}

$$P(X) = P(X_L \mid X_{L-1}) P(X_{L-1} \mid X_{L-2}) \ldots P(X_2 \mid X_1) P(X_1)$$

$$= P(X_1) \prod_{i=2}^{L} P(X_i \mid X_{i-1})$$

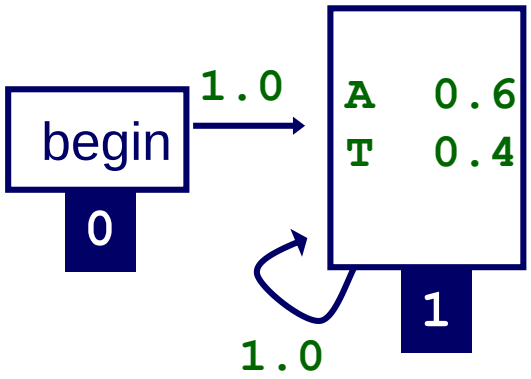
Markov chain model



The probability of a sequence cggt for a given model:

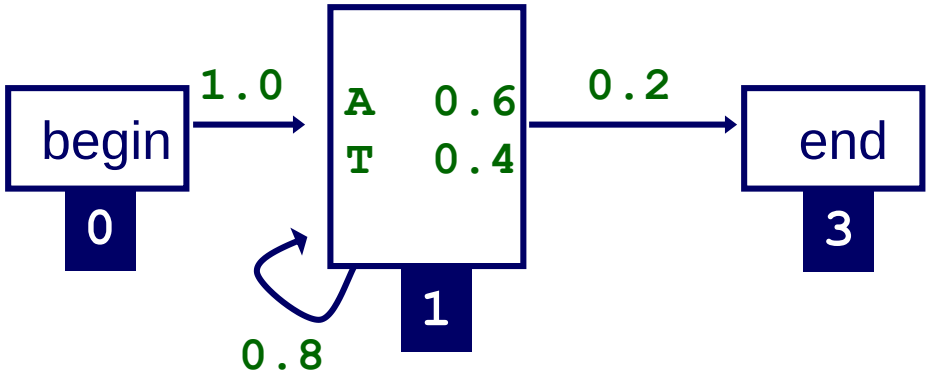
$$P(cggt) = P(c)P(g|c)P(g|g)P(t|g)P(\text{end}|t)$$

Why we need an end state to define a distribution over varying length sequences



P(A) = 0.6
P(T) = 0.4

P(AA) = 0.36
P(AT) = 0.24
P(TA) = 0.24
P(TT) = 0.16



P(A) = 0.12
P(T) = 0.08

P(AA) = 0.0576
P(AT) = 0.0384
P(TA) = 0.0384
P(TT) = 0.0256

P(L=1) = 0.2

P(L=2) = 0.16

Estimating the model parameters

- Given some sequences, how can we determine the probability parameters of our model?
  - maximum likelihood estimation
  - Bayesian approach – regularization, priors
- estimate 1st order parameters using Laplace estimates with the sequences

gccgcgcttg

gcttggtggc

tggccgttgc

$$P(a \mid g) = \frac{0+1}{12+4}$$

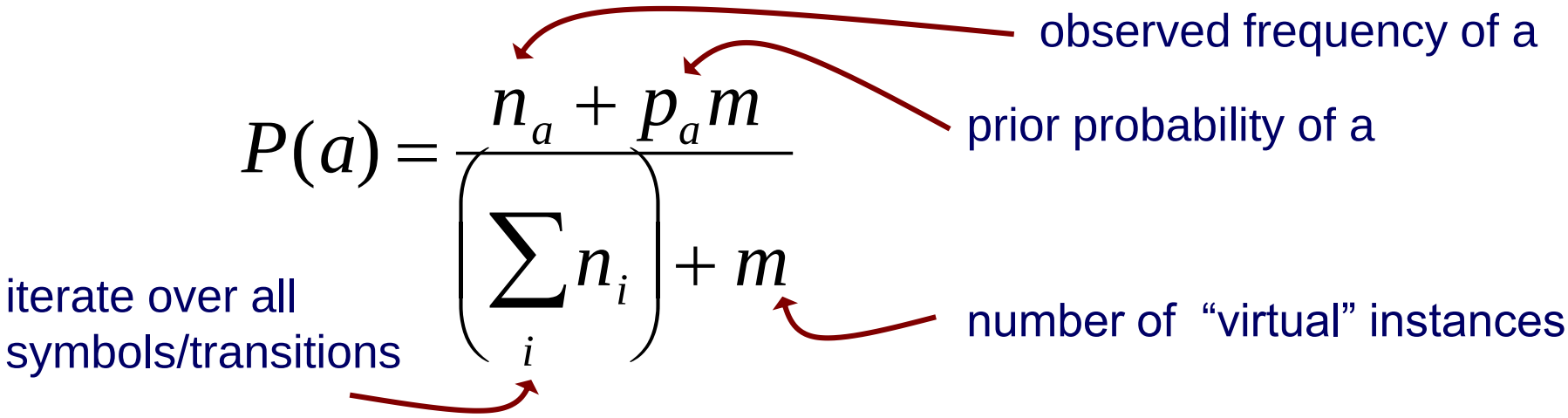
$$P(c \mid g) = \frac{7+1}{12+4}$$

$$P(g \mid g) = \frac{3+1}{12+4}$$

$$P(t \mid g) = \frac{2+1}{12+4}$$

A Bayesian approach

- instead of estimating parameters strictly from the data, we could start with some prior belief for each
- in general, use a prior Dirichlet distribution as a conjugate prior to the observed multinomial data distribution
- the outcome reduces to *m-estimates*



  - their most simple form = *Laplace estimates* (uniform p_a, $m=1/p_a$)

Higher order Markov chains

- we can build more “memory” into our states by using a higher order Markov model
- additional history can have predictive value
- example:
  - predict the next word in this sentence fragment

    “… the__” (duck, end, grain, tide, wall, …?)
  - now predict it given more history

    “… against the __” (duck, end, grain, tide, wall, …?)

    “swim against the __” (duck, end, grain, tide, wall, …?)

Higher order Markov chains

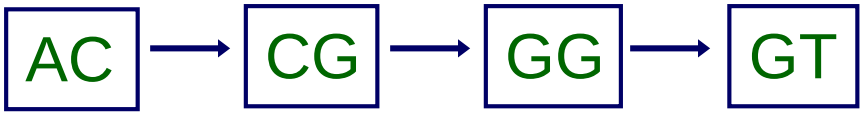
- an nth order Markov chain over some alphabet A is equivalent to a first order Markov chain over the alphabet A^n of n-tuples

- example: a 2nd order Markov model for DNA can be treated as a 1st order Markov model over alphabet

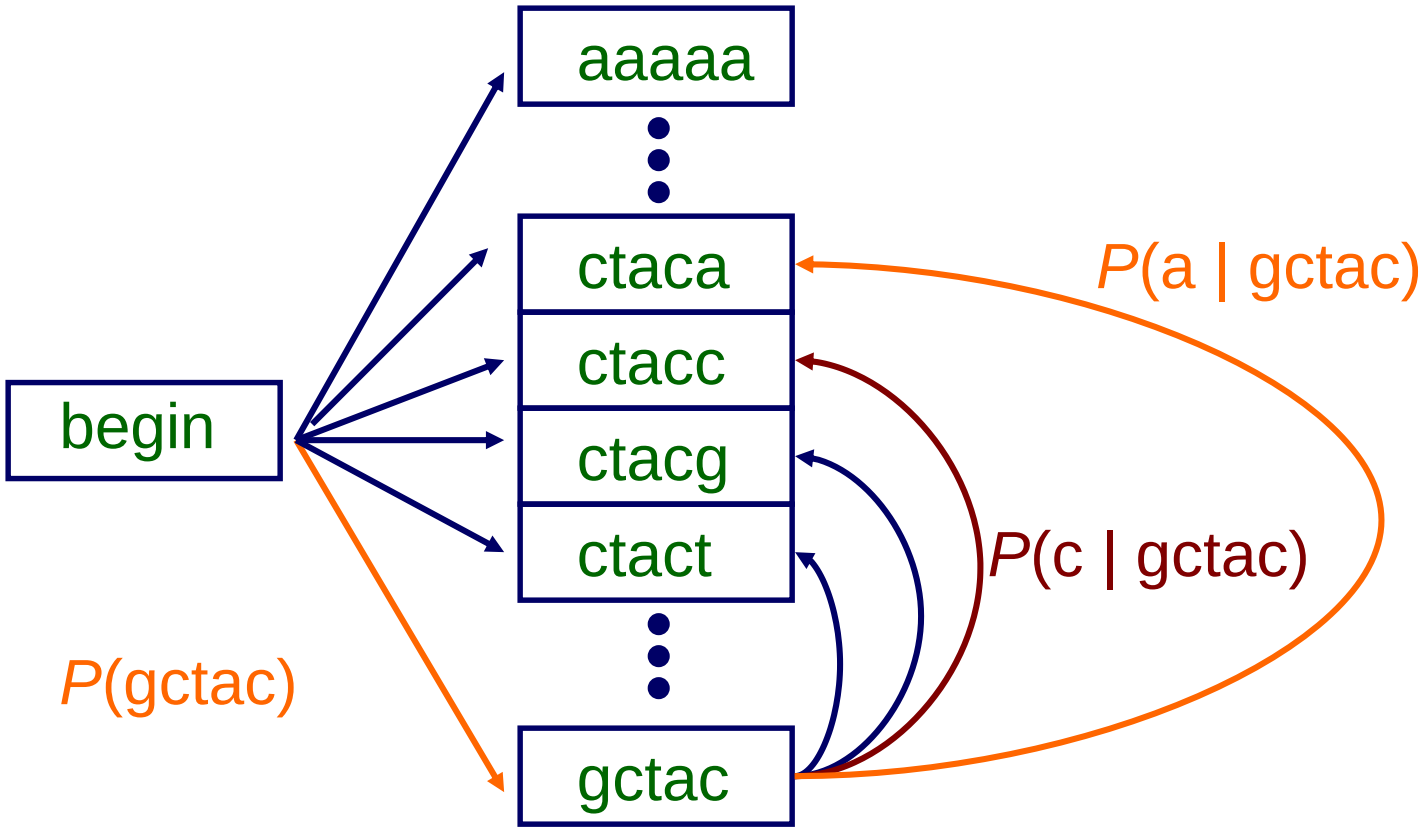
  AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT, TA, TC, TG, TT

- caveat: we process a sequence one character at a time

  A C G G T



A fifth-order Markov chain



$$P(gctaca) = P(gctac)P(a \mid gctac)$$

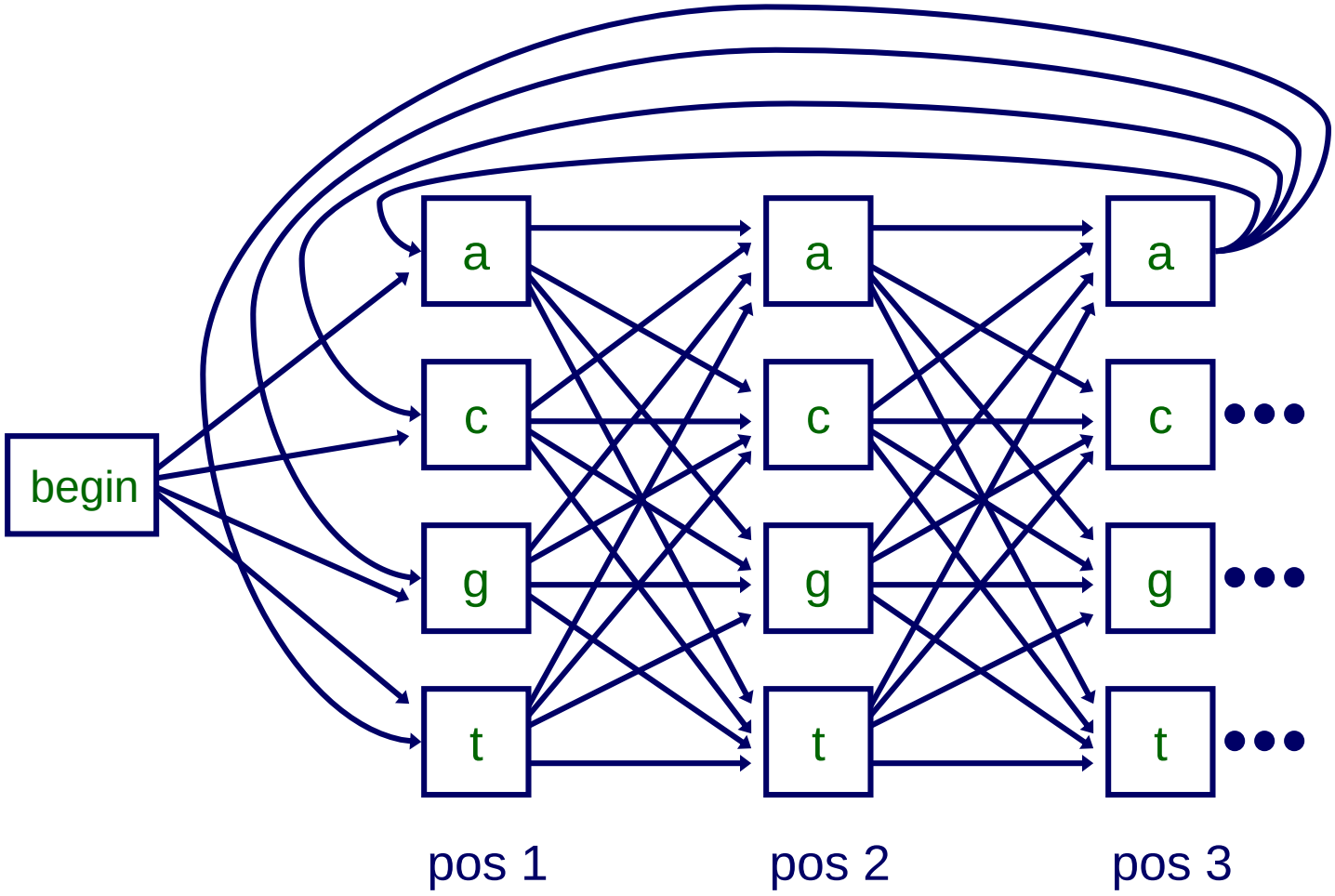
Inhomogenous Markov chains

- in the Markov chain models we have considered so far, the probabilities do not depend on our position in a given sequence

- in an *inhomogeneous* Markov model, we can have different distributions at different positions in the sequence

- consider modeling codons in protein coding regions

An inhomogeneous Markov chain



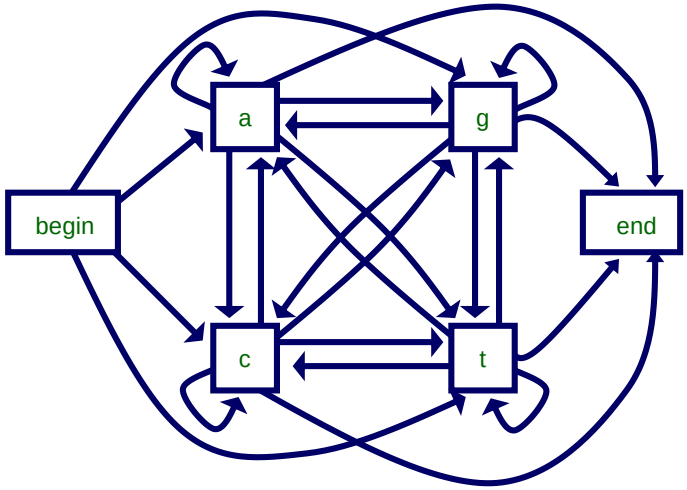
Example application

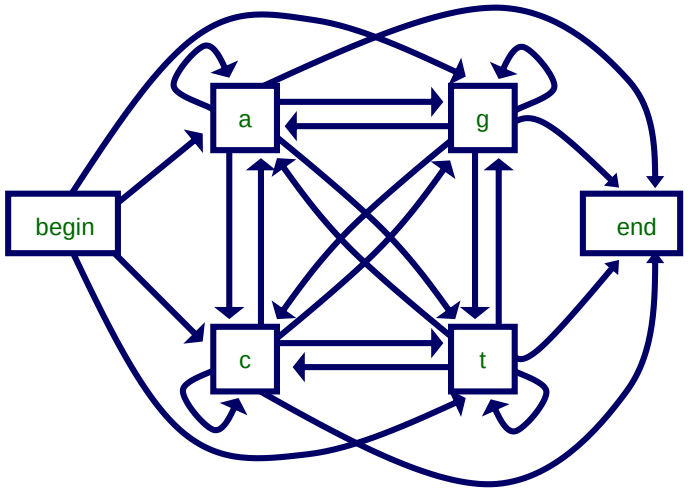
- CpG islands
  - CG dinucleotides are rarer in eukaryotic genomes than expected given the marginal probabilities of C and G
  - but the regions upstream of genes are richer in CG dinucleotides than elsewhere – *CpG islands*
  - useful evidence for finding genes

- could predict CpG islands with Markov chains
  - one to represent CpG islands
  - one to represent the rest of the genome

CpG islands as a classification task

1. train two Markov models: one to represent CpG island sequence regions, another to represent other sequence regions (*null*)





2. given a test sequence, use two models to
   - determine probability that sequence is a CpG island
   - classify the sequence (*CpG* or *null*)

Markov chains for discrimination

- parameters estimated for CpG and null models
  - human sequences containing 48 CpG islands
  - 60,000 nucleotides

$P(c \mid a)$

+	*a*	*c*	*g*	*t*
a	.18	.27	.43	.12
c	.17	.37	.27	.19
g	.16	.34	.38	.12
t	.08	.36	.38	.18

CpG

-	*a*	*c*	*g*	*t*
a	.30	.21	.28	.21
c	.32	.30	.08	.30
g	.25	.24	.30	.21
t	.18	.24	.29	.29

null

Markov chains for discrimination

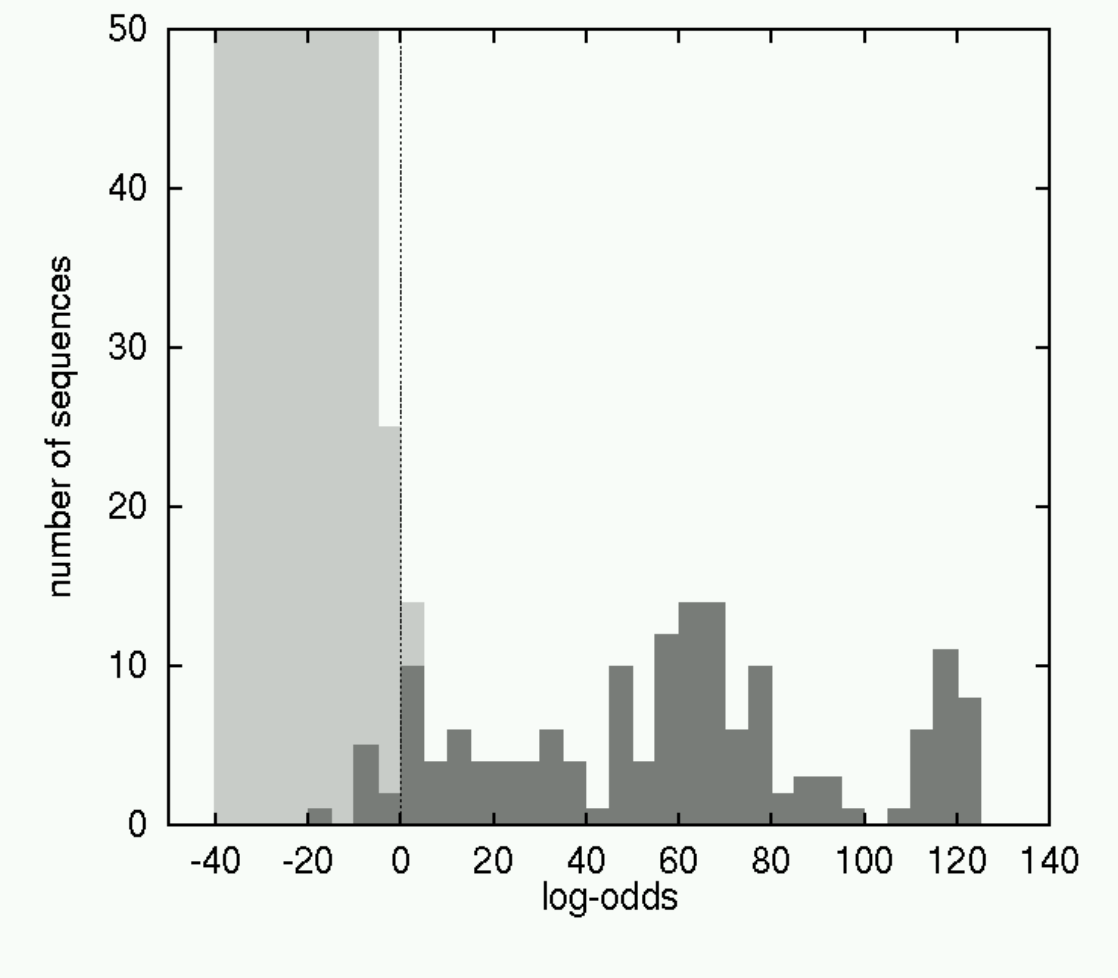
- using Bayes' rule tells us

$$P(CpG \mid x) = \frac{P(x \mid CpG)P(CpG)}{P(x)}$$

$$= \frac{P(x \mid CpG)P(CpG)}{P(x \mid CpG)P(CpG) + P(x \mid null)P(null)}$$

- if we don't take into account prior probabilities of two classes ($P(CpG)$ and $P(null)$) then we just need to compare $P(x \mid CpG)$ and $P(x \mid null)$

Markov chains for discrimination



- light bars represent negative sequences
- dark bars represent positive sequences (i.e. CpG islands)
- the actual figure here is not from a CpG island discrimination task, however

Figure from A. Krogh, “An Introduction to Hidden Markov Models for Biological Sequences” in Computational Methods in Molecular Biology, Salzberg et al. editors, 1998.

The hidden part of the problem

- in the Markov models we've considered previously, it is clear which state accounts for each part of the observed sequence
- we'll distinguish between the *observed* parts of a problem and the *hidden* parts
- in **hidden markov models**, there are multiple states that could account for each part of the observed sequence – this is the hidden part of the problem

The parameters of an HMM

- as in Markov chain models, we have transition probabilities

$$a_{kl} = P(\pi_i = l \mid \pi_{i-1} = k)$$

probability of a transition from state *k* to *l*

π represents a path (sequence of states) through the model

- since we've decoupled states and characters, we might also have emission probabilities

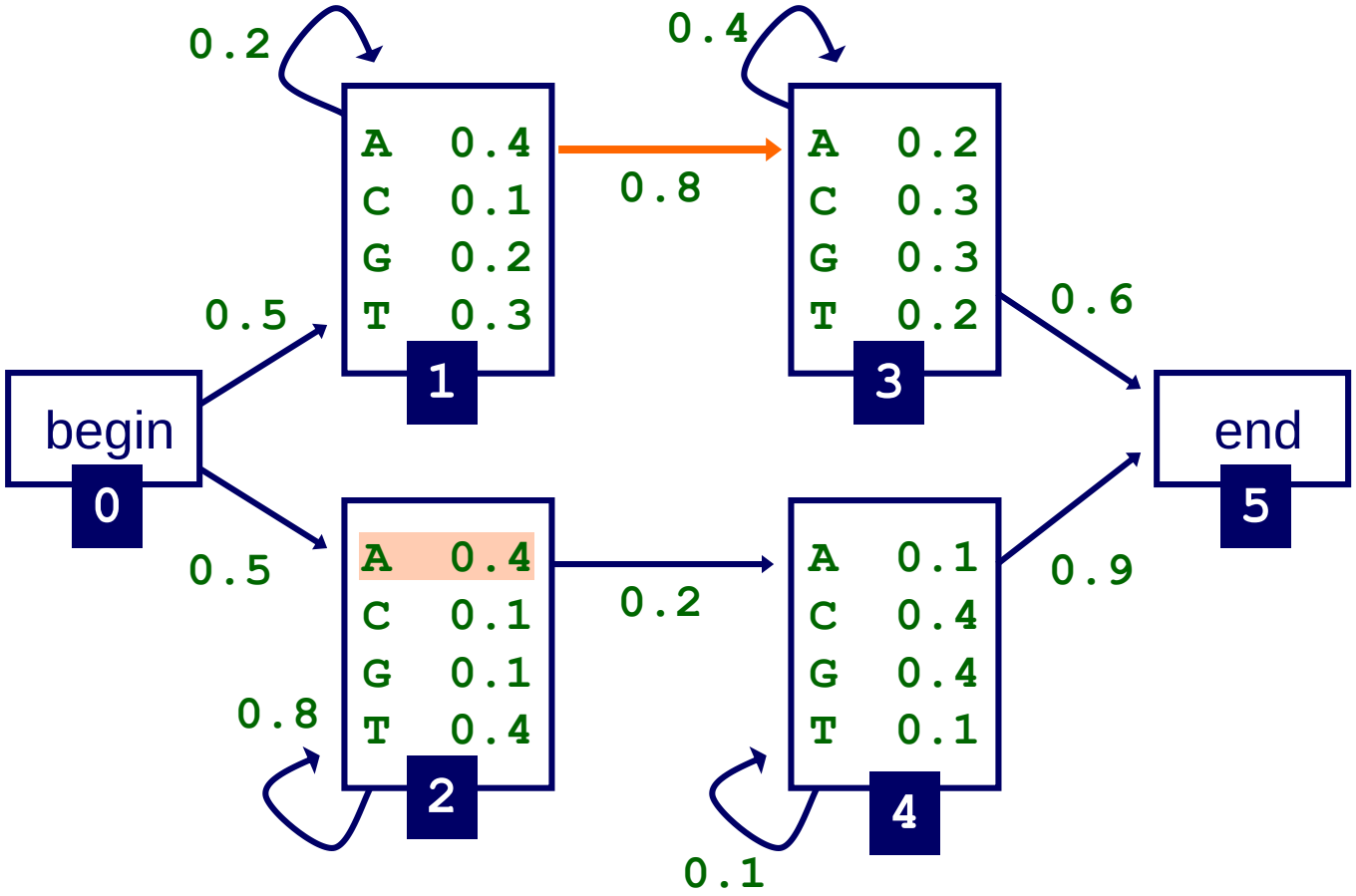
$$e_k(b) = P(x_i = b \mid \pi_i = k)$$

probability of emitting character b in state *k*

A simple HMM with emission parameters

a_{13} probability of a transition from state 1 to state 3

$e_2(\mathrm{A})$ probability of emitting character *A* in state 2



Three important HMM questions

- How likely is a given sequence given the model?
  the Forward algorithm
- What is the most probable “path” for generating a given sequence?
  the Viterbi algorithm
- How can we learn the HMM parameters given a set of sequences?
  the Forward-Backward (Baum-Welch) algorithm

Learning and prediction tasks

- *learning*

  Given: a model, a set of training sequences

  Do: find model parameters that explain the training sequences with relatively high probability (goal is to find a model that *generalizes* well to sequences we haven't seen before)

- *classification*

  Given: a set of models representing different sequence classes, a test sequence

  Do: determine which model/class best explains the sequence

- *segmentation*

  Given: a model representing different sequence classes, a test sequence

  Do: segment the sequence into subsequences, predicting the class of each subsequence