7.36/7.91 Section

CB Lectures 9 & 10

3/12/14

Biomolecular Sequence Motif

- A pattern common to a set of DNA, RNA or protein sequences that share a common biological property
  - nucleotide binding sites for a particular protein (TATA box in promoter, 5' and 3' splice site sequences)
  - amino acid residues that fold into a characteristic structure (zinc finger domains of TFs)
- Consensus sequence is the one most common occurrence of the motif (e.g. TATAAA for TATA box)
  - Stronger motifs (= more information, lower entropy, less degenerate) have less deviation from consensus sequence
- Position weight matrix gives the probability of each nucleotide or amino acid at each position
  - Assumes independence between positions
  - Can be visualized with a Sequence Logo showing probability at each position

  or with each position height scaled by the information content of that position

Shannon Entropy

- Defined over a probability distribution

$$\text{valid discrete distribution} : 0 \leq p_i \leq 1, \sum_i p_i = 1$$

$$\text{Shannon entropy (in bits)} : H(p) = -\sum_i p_i \log_2(p_i)$$

$$-\text{Note that } 0 \log_2(0) = 0$$

- The entropy measures the amount of uncertainty in the probability distribution
  - If given in bits, it's the number of bits (0/1s) you would need in order to transmit a knowledge of a state (e.g. A, C, G, or T) drawn from the probability distribution
- If there are *n* states in the distribution, what distribution has the highest entropy?

$$p_i = \frac{1}{n} \quad \forall i \qquad \Longrightarrow \qquad H(p) = \log_2(n)$$

- If there are *n* states in the distribution, what distribution has the lowest entropy?

$$p_k = 1 \quad \text{for some } k, \qquad p_i = 0 \quad \text{for } i \neq k$$

$$H(p) = 0 \quad (\text{everything is determined - no uncertainty})$$

Shannon Entropy

- Shannon entropy of position over the 4 nucleotides:

$$p_A = p_C = p_G = p_T = \frac{1}{4} \qquad H(p) = -\sum_{i=1}^{4} \frac{1}{4} \log_2 \left(\frac{1}{4}\right) = 2 \text{ bits}$$

- Information content of a position *j*: $I_j = H_{j,\text{before}} - H_{j,\text{after}}$

Generally, $H_{j,\text{ before}}$=2 bits (uniform 1/4 background composition) $I_j = 2 - H_j$

- For a motif of width *w*, if positions are independent (nucleotides at one position don't affect composition of other positions)

$$I_{\text{motif}} = \sum_{j=1}^{w} I_j = 2w - H_{\text{motif}}$$

- What's the information of a 5 nt long motif that consists only of pyrimidines (C/Ts – independent positions)? 5 bits (1 bit at each position)

Shannon Entropy

- For longer

$$p_A = p_C = p_G = p_T = \frac{1}{4} \qquad H(p) = -\sum_{i=1}^{4} \frac{1}{4} \log_2 \left(\frac{1}{4}\right) = 2 \text{ bits}$$

- Information content of a position *j*: $I_j = H_{j,\text{before}} - H_{j,\text{after}}$

Generally, $H_{j,\text{ before}}$=2 bits (uniform 1/4 background composition) $I_j = 2 - H_j$

- For a motif of width *w*, if positions are independent (nucleotides at one position don't affect composition of other positions)

$$I_{\text{motif}} = \sum_{j=1}^{w} I_j = 2w - H_{\text{motif}}$$

- What's the information of a 5 nt long motif that consists only of pyrimidines (C/Ts – independent positions)? 5 bits (1 bit at each position)

Mean-bit score of a motif

- Use relative entropy (mean bit-score) of a distribution *p* relative to the background distribution *q*

$$\sum_{k=1}^{n} p_k \log_2\left(\frac{p_k}{q_k}\right) \qquad (q_k > 0;\ \text{if } p_k = 0 \implies 0\log_2(0) = 0)$$

  - *n* is the number of states; $n=4^w$ for nucleotide sequence of width *w*
- The relative entropy is a measure of *information* of one distribution *p* relative to another *q*, not entropy/uncertainty (defined for a single distribution)
  - Better to use this for information of motif when background is non-random: For example, you have gained more information/knowledge upon observing a sequence *k* if it's rare (if $p_k<1/n$) than if it's uniformly or highly likely or ($p_k \geq 1/n$)
- For sequences with uniform background ($q_k=1/4^w$):

$$\text{Relative entropy of motif (model)} = I_{\text{motif}} = 2w - H_{\text{motif}}$$

H_{motif} is the Shannon entropy of the motif

- A motif with *m* bits of information generally occurs once every 2^m bases of random sequence

Non-random background sequences

- What is the information content of a motif (model) that consists of codons that always have the same nucleotide at the 1st and 3rd position?
  - There are 16 possible codons (4 possibilities for the first/third positions, and 4 possibilities for 2nd position).
  - Assuming these are all equally likely, p_k=1/16 for these codons, p_k=0 otherwise (e.g. for AGT codon)

$$\begin{aligned}\sum_{i=1}^{64} p_i \log_2\left(\frac{p_i}{q_i}\right) &= \sum_{k=1}^{16} p_k \log_2\left(\frac{p_k}{q_k}\right) \\ &= 16\left(\frac{1}{16}\log_2\left(\frac{1/16}{1/64}\right)\right) \\ &= \log_2(4) \\ &= 2 \text{ bits}\end{aligned}$$

- The 1st position determines the 3rd position, so the information that we've gained is complete knowledge of this position given the first position (i.e., the full information content of one position, which is 2 bits)
- Also note that the Shannon entropy of the motif is $-\sum_{i=1}^{64} p_i \log_2(p_i) = -\sum_{k=1}^{16} p_k \log_2(p_k) = -16\left(\frac{1}{16}\log_2\left(\frac{1}{16}\right)\right) = \log_2(16) = 4 \text{ bits}$

So relative to a uniform background distribution of codons (q_k=1/64), the information content of this motif is: I=2w – H_{motif} = (2*3) – 4 = 2 bits (same as calculating Relative Entropy)

Gibbs Sampler

- Type of Markov Chain Monte Carlo (MCMC) algorithm that relies on probabilistic optimization
  - Relies on repeated random sampling to obtain results
  - Due to randomness, can get different results from same starting condition; generally want to run algorithm many times and compare results to see how robust solution is
  - Determining when to stop is less well defined since random updates may or may not change at each iteration
  - Not forced to stay in local minimum; possible to "escape" it during a random sampling step
  - Initialization is less important; results from different initializations will often return similar results since they will "cross paths" at some point (sampling step)
    - Contrast this with a deterministic algorithm like the EM algorithm (GPS ChIP-seq peak-finding) – initial conditions are more important and results are deterministic given those initial conditions; cannot escape being stuck in local minimum

Gibbs Sampler

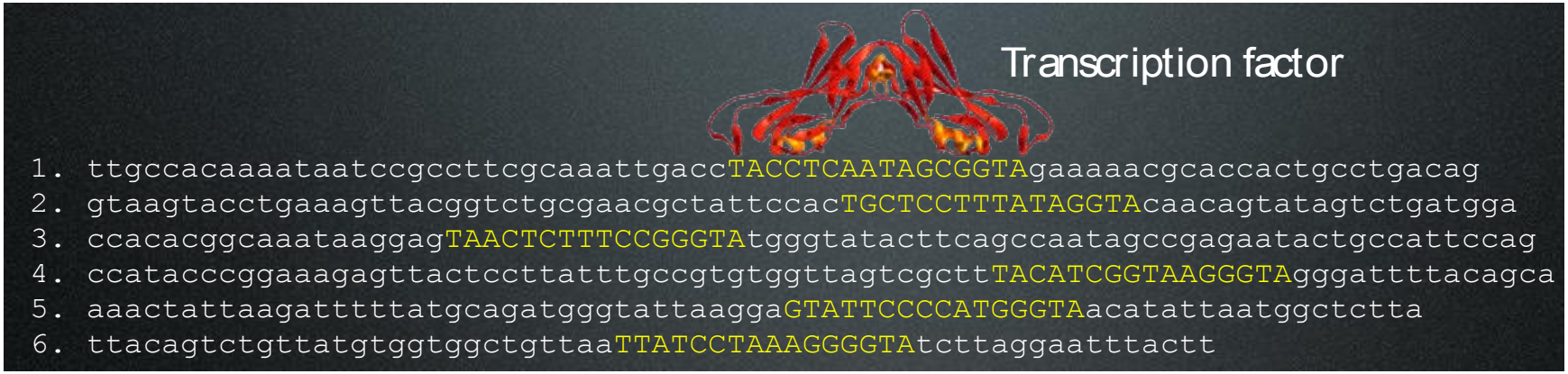
- Goal: to find a motif (position weight matrix) in a set of sequences
  - Assumption: motif is present in each of your sequences
    - Doesn't need to be true, but sequences without motif will dilute results (come up with more degenerate PWM)
- General idea: probability of a motif occurring at any position in a sequence is proportional to the probability of that subsequence being generated by the current PWM
  - Iteratively update:

    (1) PWM based on subsequences we think are motifs

    (2) Locations of the motif subsequences in the full sequences based on similarity to PWM

    -randomness comes in (2) – choosing where start of subsequence is
  - Specifically, at each step we leave one sequence out and optimize the location of the motif in the left-out sequence

Gibbs Sampler



Courtesy of Carl Kingsford. Used with permission.

Start with *N* sequences, searching for motif of length *W* (*W* < length each of sequence)

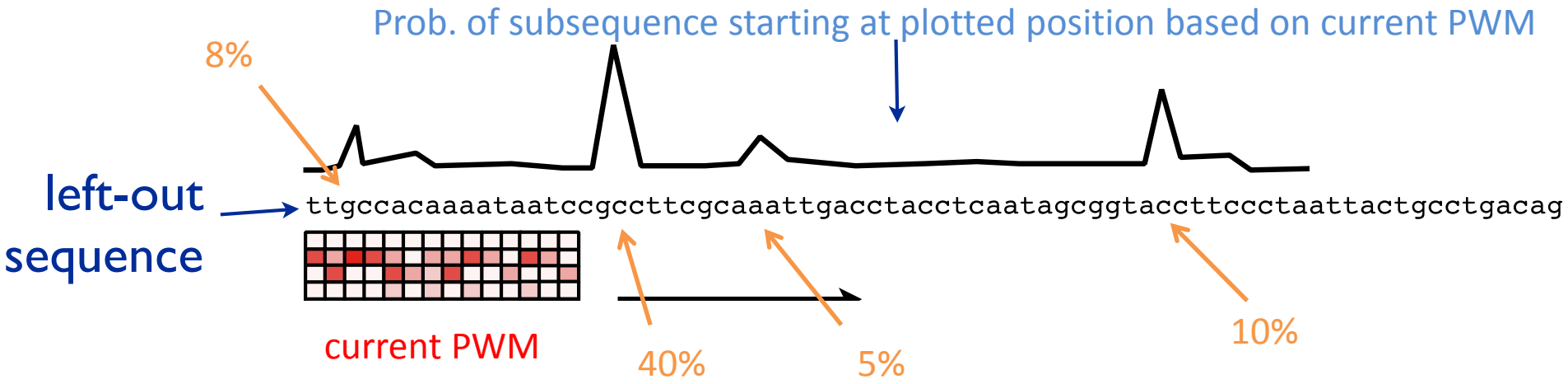
-Randomly choose a starting position in each sequence (a_1, a_2, …, a_N) – the starting guess as to where motif is in each sequence

- Randomly choose one sequence to leave-out (will optimize motif position in this sequence)

- Make a PWM from *N*-1 subsequences at the starting positions in all sequences except the one left-out

http://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/gibbs.pdf

Gibbs Sampler



Courtesy of Carl Kingsford. Used with permission.

Start with *N* sequences, searching for motif of length *W* (*W* < length each of sequence)

-Randomly choose a starting position in each sequence (a_1, a_2, …, a_N) – the starting guess as to where motif is in each sequence

- Randomly choose one sequence to leave-out (will optimize motif position in this sequence)

- Make a PWM from *N*-1 subsequences at the starting positions in all sequences except the one left-out

- For the left-out sequence (has length *L*), assign a probability from the currently estimated PWM for each of the subsequences starting at positions 1, 2, …, *L*-*W*+1

- Normalize these probabilities to sum to 1 and select one at random from that distribution. This is the new position of the motif in the left-out sequence.

http://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/gibbs.pdf

Gibbs Sampler

Courtesy of Carl Kingsford. Used with permission.

Start with *N* sequences, searching for motif of length *W* (*W* < length each of sequence)

-Randomly choose a starting position in each sequence (a_1, a_2, …, a_N) – the starting guess as to where motif is in each sequence

- Randomly choose one sequence to leave-out (will optimize motif position in this sequence)

- Make a PWM from *N*-1 subsequences at the starting positions in all sequences except the one left-out

- For the left-out sequence (has length *L*), assign a probability from the currently estimated PWM for each of the subsequences starting at positions 1, 2, …, *L*-*W*+1

- Normalize these probabilities to sum to 1 and select one at random from that distribution. This is the new position of the motif in the left-out sequence.

Repeat until convergence

http://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/gibbs.pdf

Gibbs Sampler

Courtesy of Carl Kingsford. Used with permission.

When is convergence?

- Different options: go through *N* updates (each sequence once on average) and
  - (1) No motif subsequence location changes
  - (2) PWM changes less than desired amount (e.g. 1% at each position)

-Gibbs sampler works better when:

- motif is present in each sequence
- motif is strong
- *L* is small (less flanking non-motif sequence)
- *W* is (close to) correct length

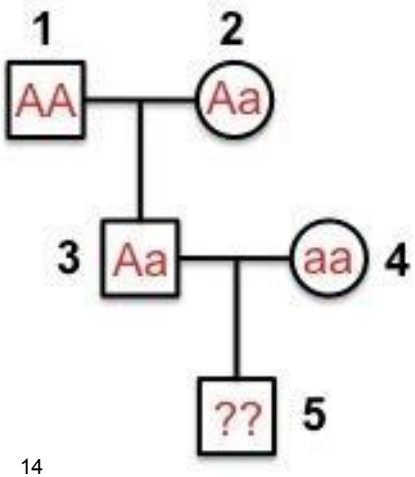
- PWM may be a shifted version of true motif (missing first or last positions of motif) – once converged, you can shift PWM by 1+ positions forward or backward and see if you converge to better results (higher likelihood of generating sequences from new PWMs)
- If computationally feasible, want to run the Gibbs sampler (1) for multiple motif lengths *W*, and (2) multiple times to gauge robustness of results

http://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/gibbs.pdf

Review: Markov Chains

- Defined by a set of *n* possible states s_1, ..., s_n at each timepoint
- Markov models follow the **Markov Property**: Transition from state *i* to *j* (with probability $P_{i,j}$) depends *only* on the previous state, not any states before that. In other words, the future is conditionally independent of the past given the present:

$$P(S_{t+1} = k | S_1 = s_1, ..., S_t = s_t) = P(S_{t+1} = k | S_t = s_t)$$



Example: if we know individual 3's genotype, there's no additional information that individuals 1 and 2 can give us about 5's genotype. So current state (individual 5's genotype) depends only on previous state (individuals 3 and 4).

Hidden Markov Models

- What if we cannot observe the states (genotypes) directly, but instead observe some phenotype that depends on the state

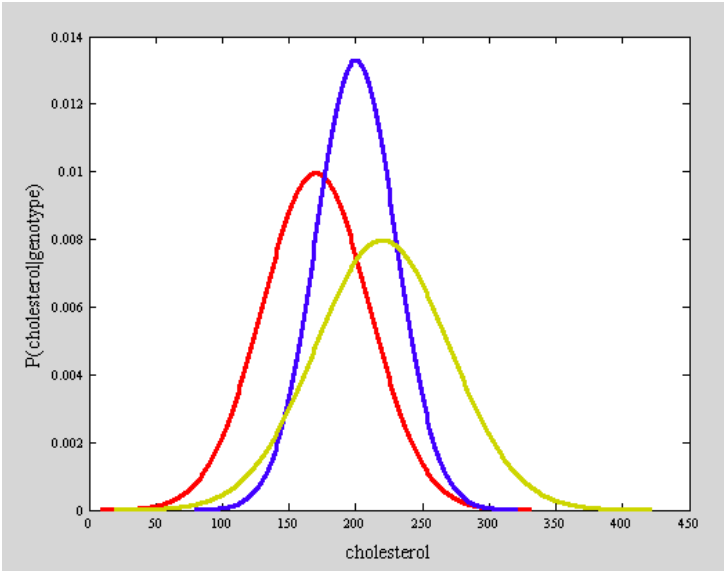
Actual genotypes unknown

1 $x_1 = 210$

3 $x_3 = 205$

5 $x_5 = 160$

Instead we observe cholesterol level x, which is distributed differently for different genotypes G:



$X \sim \mathcal{N}(170, 40)$ if genotype is aa

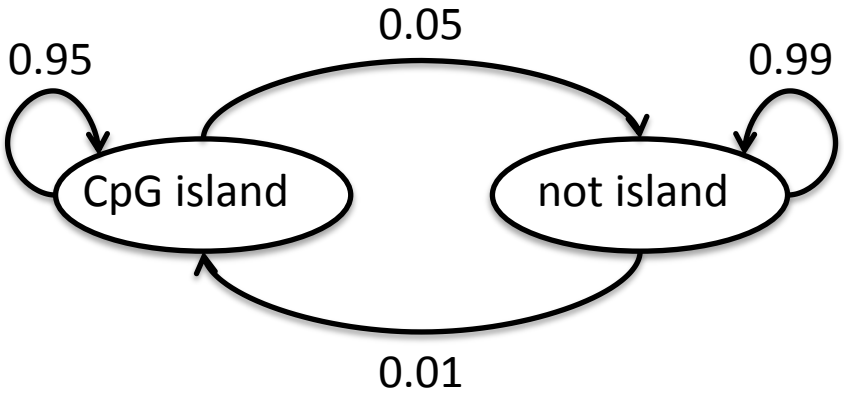
$X \sim \mathcal{N}(200, 30)$ if genotype is Aa

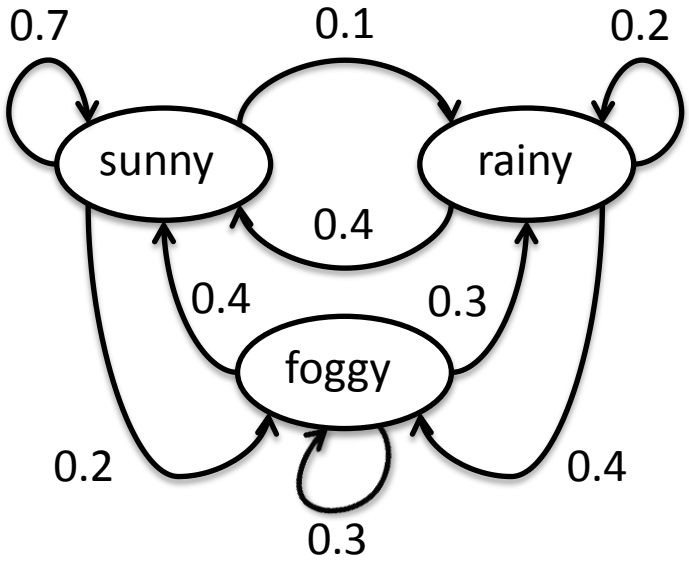
$X \sim \mathcal{N}(220, 50)$ if genotype is AA

This is now a Hidden Markov Model – and we want to infer the most likely sequence of hidden genotypes for individuals 1,3, and 5, given observed cholesterol levels, and transition probabilities between genotypes.

Graphical Representations

- can be represented graphically by drawing circles for states, and arrows to indicate transitions between states





arrow weights indicate probability of that transition

each hidden state "emits" an observable variable whose distribution depends on the state – what can we actually observe from the CpG island model?

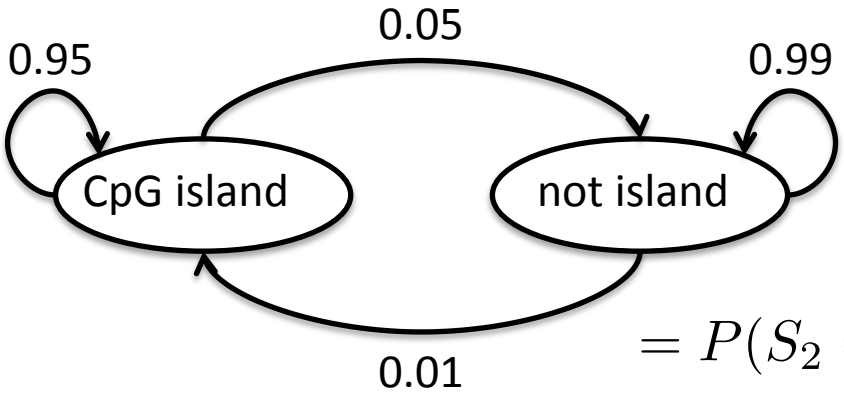
we observe the bases A, T, G, C, where observing a G or C is more likely in a CpG island

what might we observe to infer the state in this "weather" model?
(pretend you can't see the weather because you're toiling away in a basement lab with no windows)

we could use whether or not people brought their umbrellas to lab

Graphical Representations

- can be represented graphically by drawing circles for states, and arrows to indicate transitions between states



Given that we're currently in a CpG island, what is the probability that the next two states are CpG island (I) and not island (G), respectively?

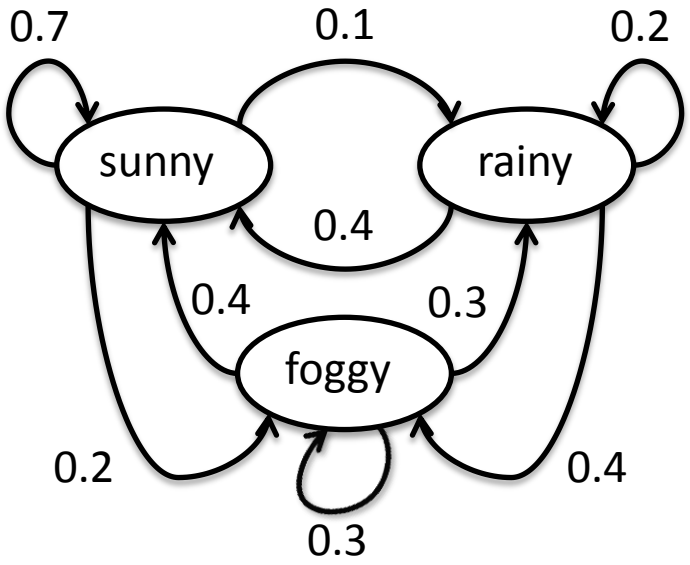
$$P(S_2 = I, S_3 = G | S_1 = I)$$

$$= P(S_2 = I | S_1 = I) * P(S_3 = G | S_1 = I, S_2 = I)$$

markov property →

$$= P(S_2 = I | S_1 = I) * P(S_3 = G | S_2 = I)$$

$$= 0.95 * 0.05 = 0.0475$$



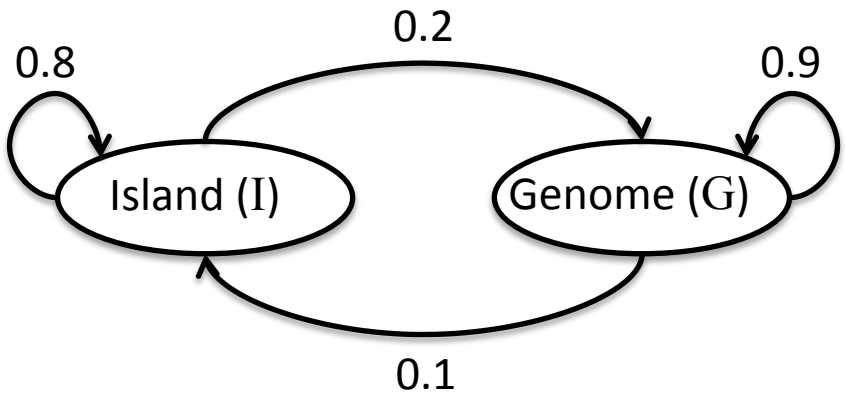
If it's currently rainy, what's the probability that it will be rainy 2 days from now?

$$P(S_3 = R | S_1 = R)$$

Need to sum the probabilities over the 3 possible paths RRR, RSR, RFR:

$$= (0.2)(0.2) + (0.4)(0.1) + (0.4)(0.3) = 0.2$$

HMMs continued



What information do we need in order to fully specify one of these models?

(1) $P_1(S)$ = probability of starting in a particular state S (vector with dimension = # of states)

$$P_1(S) = \begin{matrix} \text{I} & \text{G} \end{matrix} \begin{bmatrix} 0.1 & 0.9 \end{bmatrix}$$

(2) probability of transitioning from one state to another (square matrix w/ each dimension = # of states, usually called the transition matrix, T)

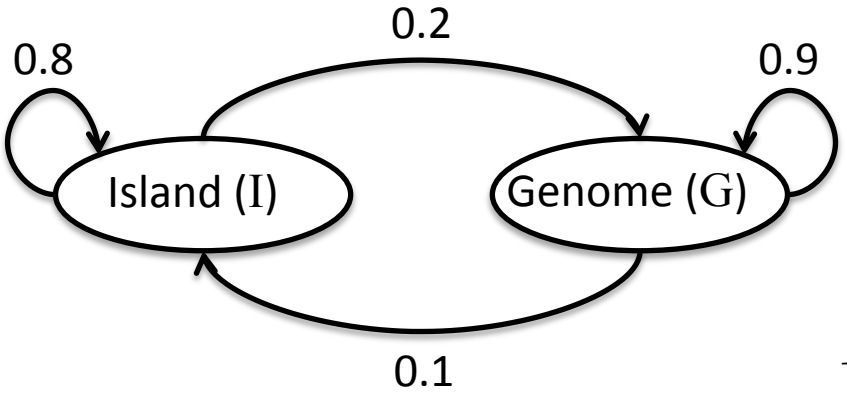
$$T = \begin{matrix} \\ \text{I} \\ \text{G} \end{matrix} \begin{matrix} \begin{matrix} \text{I} & \text{G} \end{matrix} \\ \begin{bmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{bmatrix} \end{matrix}$$

(3) $P_{\text{E}}(X|S)$ = probability of emitting X given current state S

$$P(X|S = I) = \begin{matrix} \text{C} & \text{G} & \text{A} & \text{T} \end{matrix} \begin{bmatrix} 0.4 & 0.4 & 0.1 & 0.1 \end{bmatrix}$$

$$P(X|S = G) = \begin{bmatrix} 0.1 & 0.1 & 0.4 & 0.4 \end{bmatrix}$$

Using HMMs as generative models



$$P_1(S) = \begin{matrix} \text{I} & \text{G} \\ \end{matrix} \begin{bmatrix} 0.1 & 0.9 \end{bmatrix}$$

$$P(X|S=I) = \begin{matrix} \text{C} & \text{G} & \text{A} & \text{T} \\ \end{matrix} \begin{bmatrix} 0.4 & 0.4 & 0.1 & 0.1 \end{bmatrix}$$

$$P(X|S=G) = \begin{bmatrix} 0.1 & 0.1 & 0.4 & 0.4 \end{bmatrix}$$

We want to generate a DNA sequence of length L that could be observed from this model

(1) choose initial state from $P_1(S)$

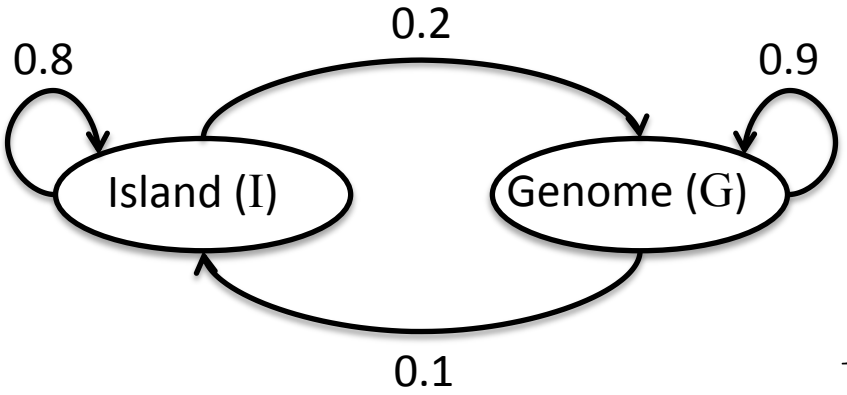
(2) emit first base of sequence according to current state and $P_{\text{E}}(X|S)$

for $1 < i < L$:

(3) choose state at position i according to transition matrix and state at position $i - 1$, e.g. using $P_{\text{T}}(S_i|S_{i-1})$

(4) emit base of sequence according to current state S_i and $P_{\text{E}}(X|S_i)$

The Viterbi Algorithm



$$P_1(S) = \begin{matrix} \text{I} & \text{G} \\ \end{matrix} \begin{bmatrix} 0.1 & 0.9 \end{bmatrix}$$

$$P(X|S=I) = \begin{matrix} \text{C} & \text{G} & \text{A} & \text{T} \\ \end{matrix} \begin{bmatrix} 0.4 & 0.4 & 0.1 & 0.1 \end{bmatrix}$$

$$P(X|S=G) = \begin{bmatrix} 0.1 & 0.1 & 0.4 & 0.4 \end{bmatrix}$$

Often, we want to infer the most likely sequence of hidden states S for a particular sequence of observed values O (e.g. bases); in other words, find

$S^{opt} = s_1^{opt}, s_2^{opt}, \ldots$ that maximizes $P(S = s_1, \ldots, s_n, O = o_1, \ldots, o_n)$

-what is the optimal parse for the following sequence? GTGCCTA

-we're going to find this recursively, e.g. we find optimal parse of the first two bases GT in terms of paths up to the first base, G

What is the optimal parse for the first base, G?

- if first state is I?

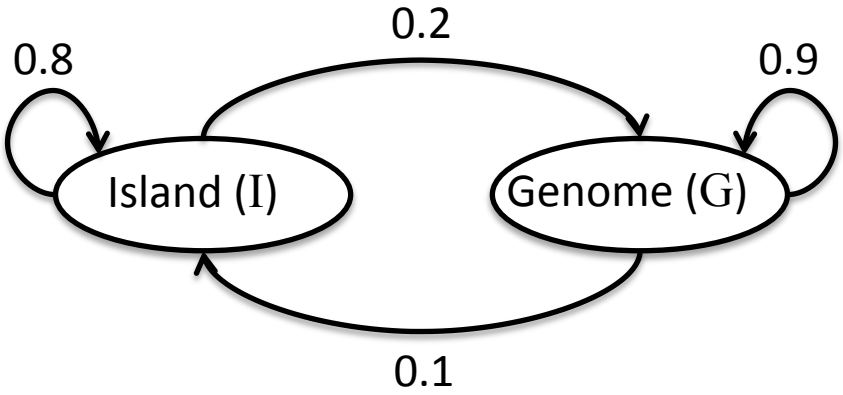
$P(X_1 = \text{G} \mid S_1 = \text{I}) = P_1(\text{I}) * P_E(\text{G} \mid \text{S=I}) = (0.1)*(0.4) = 0.04$

- if first state is G?

$P(X_1 = \text{G} \mid S_1 = \text{G}) = P_1(\text{G}) * P_E(\text{G} \mid \text{S=G}) = (0.9)*(0.1) = 0.09$

Therefore, the optimal parse for the first base is state G (note this doesn't yet consider the rest of the sequence!)

Using HMMs as generative models

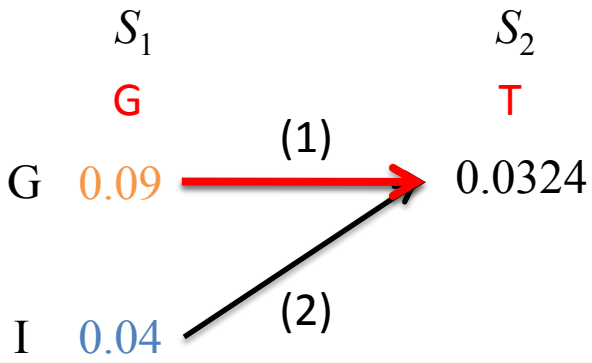


$$P_1(S) = \begin{matrix} \text{I} & \text{G} \\ \end{matrix} \begin{bmatrix} 0.1 & 0.9 \end{bmatrix}$$

$$P(X|S=I) = \begin{matrix} \text{C} & \text{G} & \text{A} & \text{T} \\ \end{matrix} \begin{bmatrix} 0.4 & 0.4 & 0.1 & 0.1 \end{bmatrix}$$

$$P(X|S=G) = \begin{bmatrix} 0.1 & 0.1 & 0.4 & 0.4 \end{bmatrix}$$

What is the most likely parse for the following sequence? GTGCCTA



Two possible ways of being in state G in position 2:

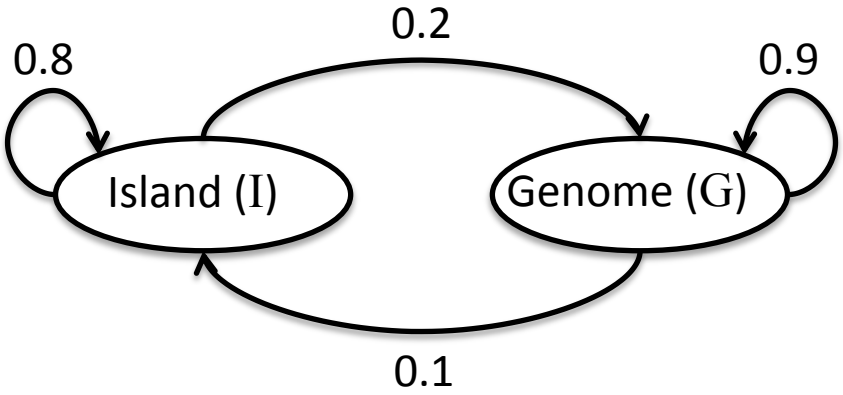
prob of optimal sequence of hidden states ending with state G at pos. 1

(1) $S_1 = \text{G}$: $P(S_1, S_2, X_1, X_2) = 0.09 * P_T(\text{G} \mid \text{G}) * P_E(\text{T} \mid \text{G})$
$= 0.09 * 0.9 * 0.4 = 0.0324$

prob of optimal sequence of hidden states ending with state I at pos. 1

(2) $S_1 = \text{I}$: $P(S_1, S_2, X_1, X_2) = 0.04 * P_T(\text{G} \mid \text{I}) * P_E(\text{T} \mid \text{G})$
$= 0.04 * 0.2 * 0.4 = 0.0032$

Using HMMs as generative models



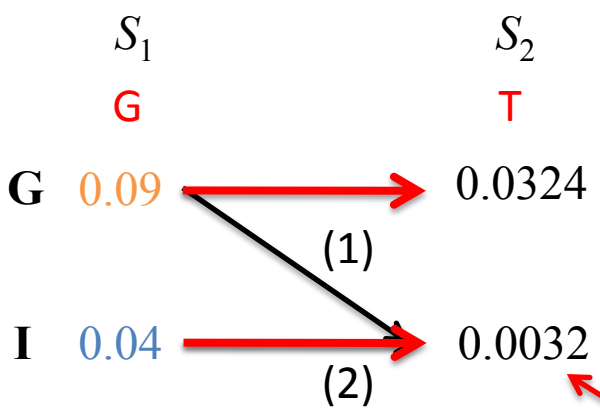
$$P_1(S) = \begin{matrix} \text{I} & \text{G} \\ [0.1 & 0.9] \end{matrix}$$

$$P(X|S=I) = \begin{matrix} \text{C} & \text{G} & \text{A} & \text{T} \\ [0.4 & 0.4 & 0.1 & 0.1] \end{matrix}$$

$$P(X|S=G) = [0.1 \quad 0.1 \quad 0.4 \quad 0.4]$$

What is the most likely parse for the following sequence? GTGCCTA

Now consider the possible ways of being in state I in position 2:

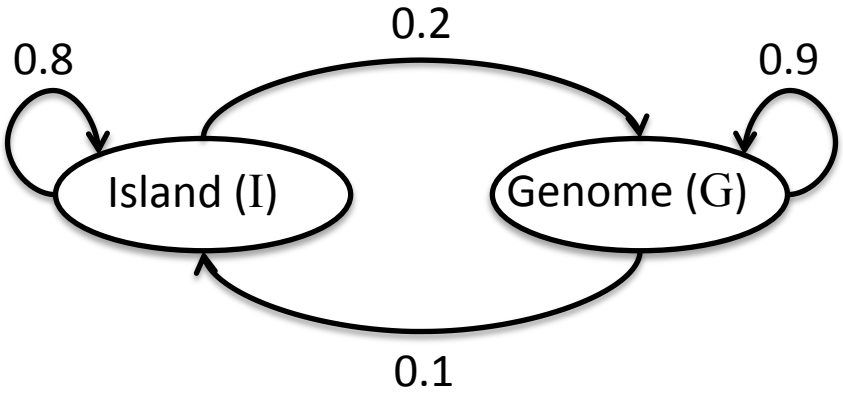


(1) S_1 = G: $P(S_1, S_2, X_1, X_2) = 0.09 * P_T(I \mid G) * P_E(T \mid I)$
$= 0.09 * 0.1 * 0.1 = 0.0009$

(2) S_1 = I: $P(S_1, S_2, X_1, X_2) = 0.04 * P_T(I \mid I) * P_E(T \mid I)$
$= 0.04 * 0.8 * 0.1 = 0.0032$

probability of the optimal parse *ending with state I at position 2 is* $\{S_1=I, S_2=I\}$

Using HMMs as generative models

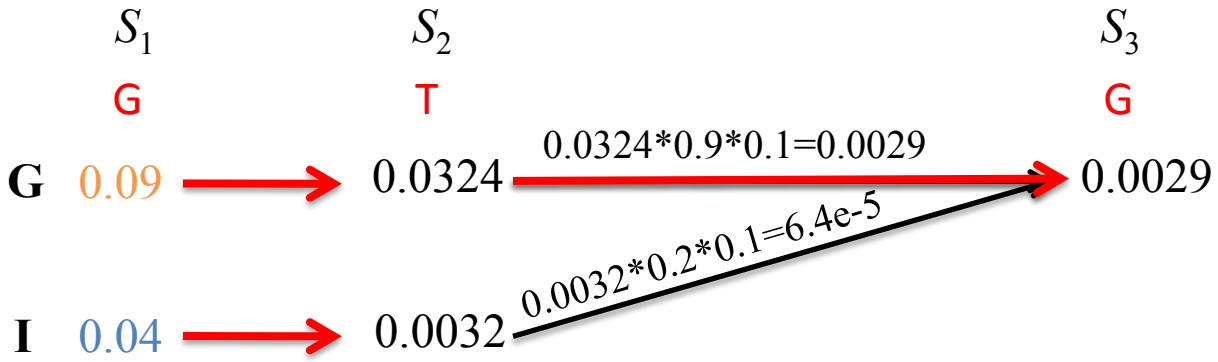


$$P_1(S) = \begin{matrix} \text{I} & \text{G} \\ \end{matrix} \begin{bmatrix} 0.1 & 0.9 \end{bmatrix}$$

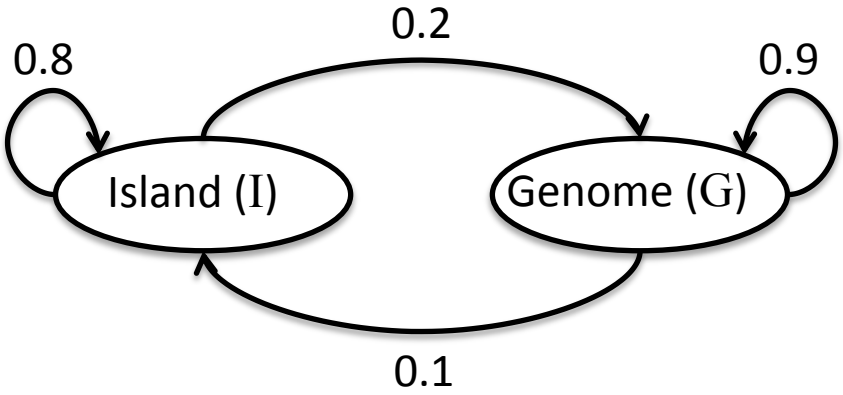
$$P(X|S=I) = \begin{matrix} \text{C} & \text{G} & \text{A} & \text{T} \\ \end{matrix} \begin{bmatrix} 0.4 & 0.4 & 0.1 & 0.1 \end{bmatrix}$$

$$P(X|S=G) = \begin{bmatrix} 0.1 & 0.1 & 0.4 & 0.4 \end{bmatrix}$$

What is the most likely parse for the following sequence? GTGCCTA



Using HMMs as generative models

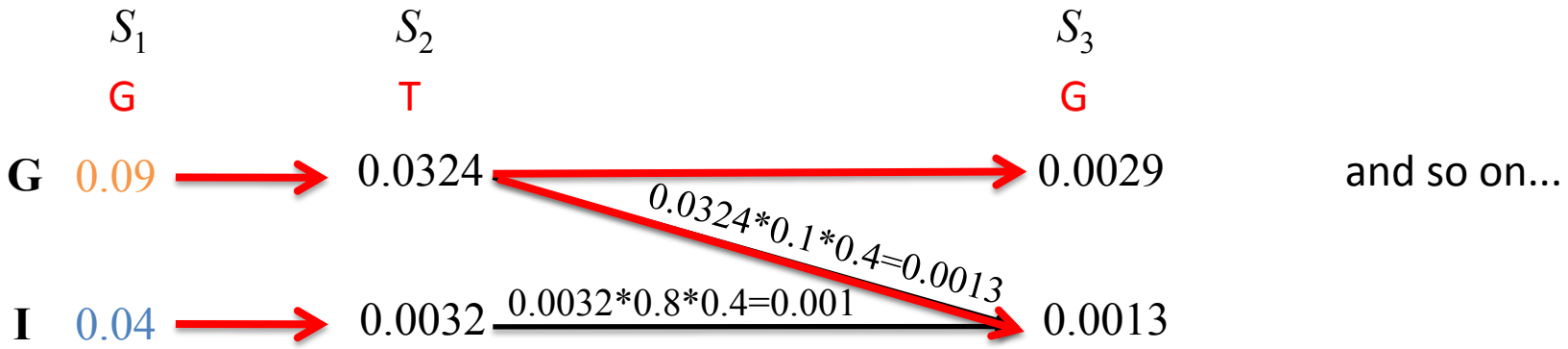


$$P_1(S) = \begin{matrix} \text{I} & \text{G} \\ \end{matrix} \begin{bmatrix} 0.1 & 0.9 \end{bmatrix}$$

$$P(X|S=I) = \begin{matrix} \text{C} & \text{G} & \text{A} & \text{T} \\ \end{matrix} \begin{bmatrix} 0.4 & 0.4 & 0.1 & 0.1 \end{bmatrix}$$

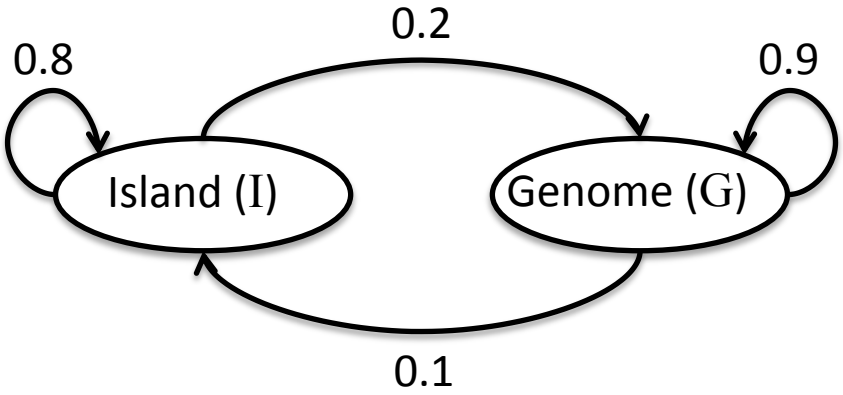
$$P(X|S=G) = \begin{bmatrix} 0.1 & 0.1 & 0.4 & 0.4 \end{bmatrix}$$

What is the most likely parse for the following sequence? GTGCCTA



and so on...

Using HMMs as generative models



$$P_1(S) = \begin{matrix} \text{I} & \text{G} \\ \end{matrix} \begin{bmatrix} 0.1 & 0.9 \end{bmatrix}$$

$$P(X|S=I) = \begin{matrix} \text{C} & \text{G} & \text{A} & \text{T} \\ \end{matrix} \begin{bmatrix} 0.4 & 0.4 & 0.1 & 0.1 \end{bmatrix}$$

$$P(X|S=G) = \begin{bmatrix} 0.1 & 0.1 & 0.4 & 0.4 \end{bmatrix}$$

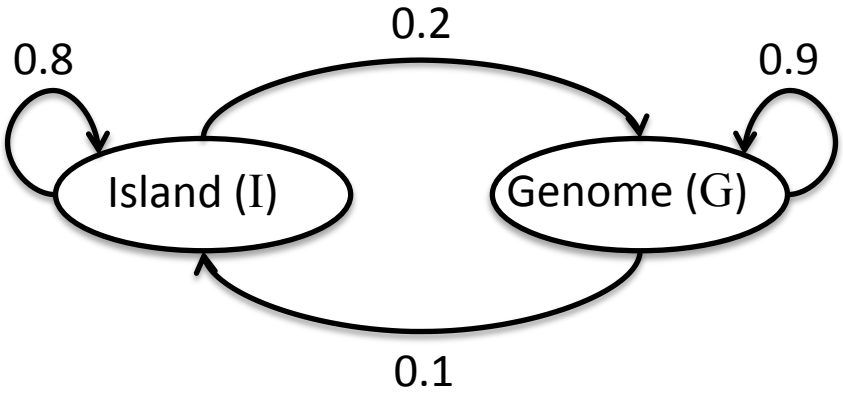
What is the most likely parse for the following sequence? GTGCCTA

	S_1	S_2	S_3	S_4	S_5	S_6	S_7
	G	T	G	C	C	T	A
G	0.09 →	0.0324 →	0.0029 →	2.61e-4 →	2.35e-5	1.06e-5 →	3.83e-6
I	0.04 →	0.0032	0.0013 →	4.16e-4 →	1.33e-4	1.06e-5 →	8.52e-7

Starting from highest final probability, traceback the path of hidden states:

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

Using HMMs as generative models



$$P_1(S) = \begin{matrix} \text{I} & \text{G} \\ \end{matrix} \begin{bmatrix} 0.1 & 0.9 \end{bmatrix}$$

$$P(X|S=I) = \begin{matrix} \text{C} & \text{G} & \text{A} & \text{T} \\ \end{matrix} \begin{bmatrix} 0.4 & 0.4 & 0.1 & 0.1 \end{bmatrix}$$

$$P(X|S=G) = \begin{bmatrix} 0.1 & 0.1 & 0.4 & 0.4 \end{bmatrix}$$

What is the most likely parse for the following sequence? GTGCCTA

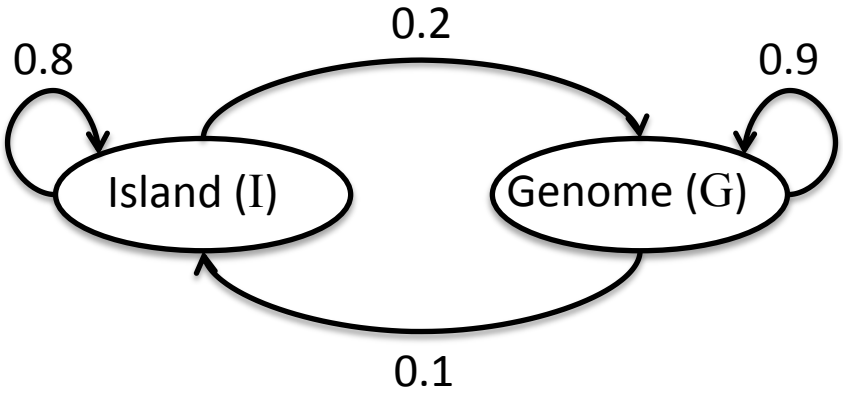
	S_1	S_2	S_3	S_4	S_5	S_6	S_7
	G	T	G	C	C	T	A
G	0.09	0.0324	0.0029	2.61e-4	2.35e-5	1.06e-5	3.83e-6
I	0.04	0.0032	0.0013	4.16e-4	1.33e-4	1.06e-5	8.52e-7

How many possible paths do we consider when advancing one position (from L-1 to L)?

$$\sum_{i=1}^{k} \sum_{j=1}^{k} P(\text{best path ending in } S_i \text{ at } L-1) * P(\text{transition from } S_i \rightarrow S_j \text{ and emit from } S_j \text{ at } L)$$

Answer: k^2. Therefore the run-time to obtain the optimal path up through pos. L is $O(k^2L)$.

Using HMMs as generative models



$$P_1(S) = \begin{matrix} \text{I} & \text{G} \\ \end{matrix} \begin{bmatrix} 0.1 & 0.9 \end{bmatrix}$$

$$P(X|S=I) = \begin{matrix} \text{C} & \text{G} & \text{A} & \text{T} \\ \end{matrix} \begin{bmatrix} 0.4 & 0.4 & 0.1 & 0.1 \end{bmatrix}$$

$$P(X|S=G) = \begin{bmatrix} 0.1 & 0.1 & 0.4 & 0.4 \end{bmatrix}$$

What is the most likely parse for the following sequence? GTGCCTA

	S_1	S_2	S_3	S_4	S_5	S_6	S_7
	G	T	G	C	C	T	A
G	0.09	0.0324	0.0029	2.61e-4	2.35e-5	1.06e-5	3.83e-6
I	0.04	0.0032	0.0013	4.16e-4	1.33e-4	1.06e-5	8.52e-7

What is optimal parse of the first 3 bases GTG?

G G G

G T G

We start at the highest probability for the last base, so we begin our traceback from the circled point above

Midterm topics

R Feb 06 CB L2 DNA Sequencing Technologies, Local Alignment (BLAST) and Statistics

T Feb 11 CB L3 Global Alignment of Protein Sequences

R Feb 13 CB L4 Comparative Genomic Analysis of Gene Regulation

R Feb 20 DG L5 Library complexity and BWT

T Feb 25 DG L6 Genome assembly

R Feb 27 DG L7 ChIP-Seq analysis (DNA-protein interactions)

T Mar 04 DG L8 RNA-seq analysis (expression, isoforms)

R Mar 06 CB L9 Modeling & Discovery of Sequence Motifs

T Mar 11 CB L10 Markov & Hidden Markov Models (+HMM content on 3/13)

7.91J / 20.490J / 20.390J / 7.36J / 6.802 / 6.874 / HST.506 Foundations of Computational and Systems Biology
Spring 2014