

# Hidden Markov Models

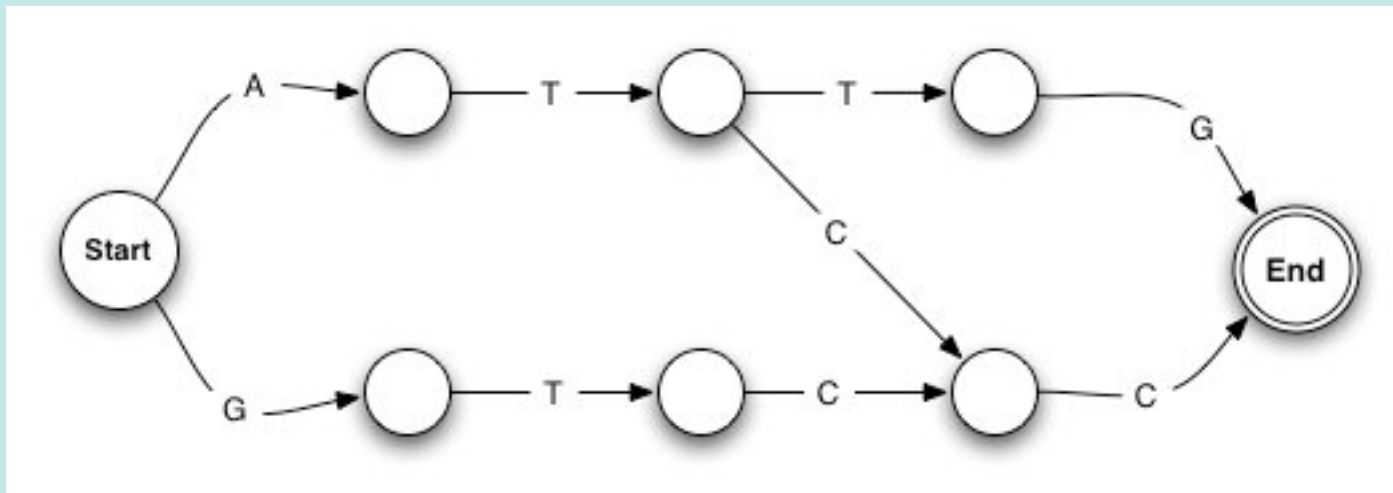
Scott Christley

Dept. of Computer Science and  
Engineering

University of Notre Dame

# Finite State Automata

- Nodes are states; edges are transitions
- Special states: Start, End
- The automata “runs” by reading the input one symbol at a time and moving from state to state; match input symbol to edge label.
- This automata recognizes the strings:
  - ATTG, ATCC, GTCC
  - GACC -- invalid!
- Automata can emit symbols instead; all paths are taken to produce strings.
- Would like to represent the difference frequency of occurrence.



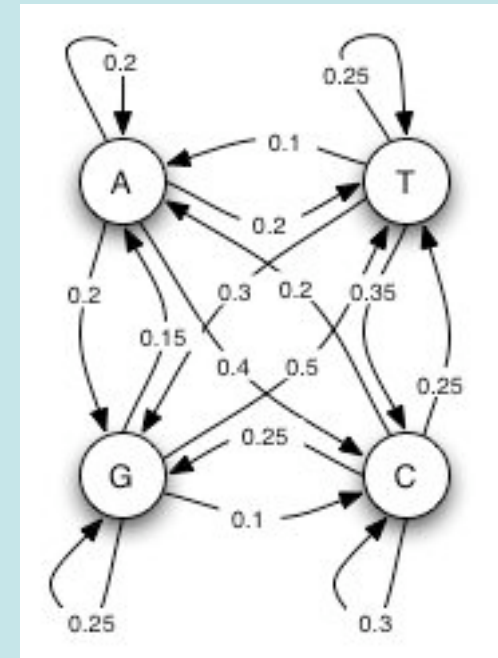
# Markov Chain

- Probabilities on the edges and symbols are moved to states.
- Probability of sequence  $x$ :

$$P(x) = P(x_1) \prod_{i=2}^L a_{x_{i-1}x_i}$$

where  $a_{x_{i-1}x_i}$  are transition probabilities

- Next state is only dependent upon current state.
- The transition probabilities can be calculated from a set of sequences by counting the number of transitions from one symbol to another.
- Different chains can be built from different sets of sequences; then each chain can be used as a discriminator for unknown sequences.
- Emit strings: sequences occur at different frequencies based upon their probability.

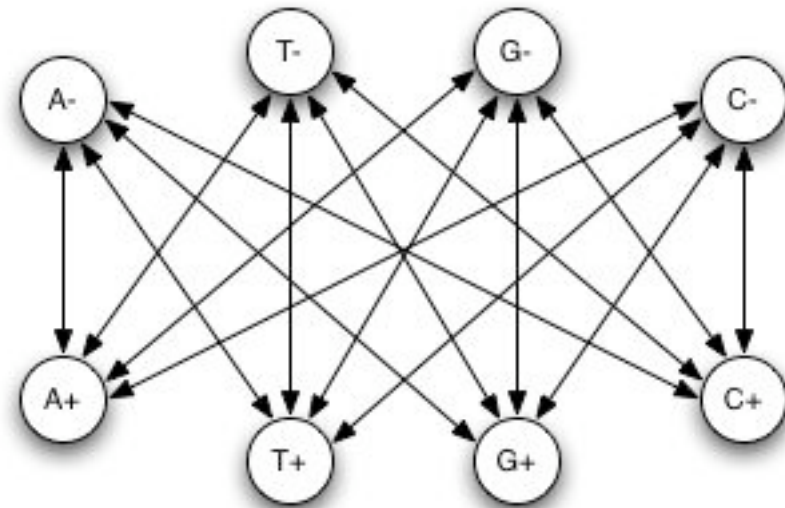


# CpG islands

- CG is the least frequent dinucleotide.
- C is typically chemically modified by methylation with the result that it has a relatively high chance to mutate into a T.
- Methylation is suppressed in regions of the genome called CpG islands, and these regions have a higher percentage of C's and G's than elsewhere in the genome.
- Questions:
  - Given a short sequence, can we decide if it comes from a CpG island?
  - Given a long sequence, can we find the CpG islands within it?

# Markov Chain -> Hidden Markov Model

- First question answered by plugging sequence into two chains, one for CpG island and another for not, calculate and compare the probabilities.
- Second question is harder. Would have to chop up the sequence, but of what length and where to chop? CpG islands presumably have crisp boundaries and are of variable length.
- Combine the two chains into one and relabel the states.
- Each group still has their own transition probabilities; probability to switch from one group to the other.
- No longer a one-to-one correspondence between states and symbols.
- Emit strings: don't know what state produced them...the states are "hidden".



# Hidden Markov Model

- Emission probabilities for symbols at each state.
- The probability of a DNA sequence  $x$  and state sequence  $\pi$  :

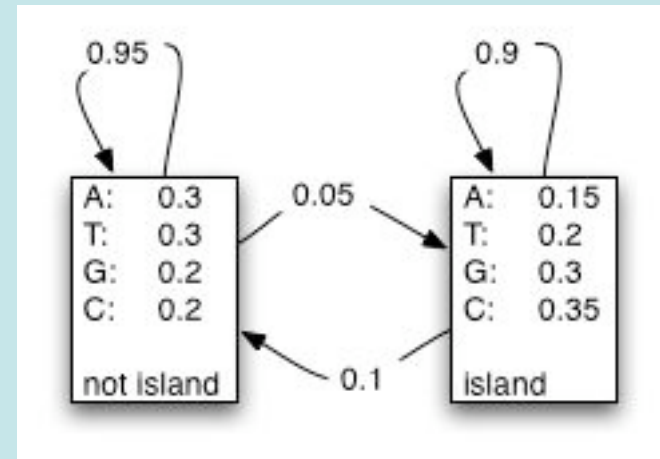
$$P(x, \pi) = a_{0\pi_1} \prod_{i=1}^L e_{\pi_i}(x_i) a_{\pi_i\pi_{i+1}}$$

where  $a_{\pi_i\pi_{i+1}}$  are transition probabilities

and  $e_{\pi_i}(x_i)$  are emission probabilities

- Many different state sequences can give rise to the same DNA sequence, each with different probability.
- Most probable state sequence:

$$\pi^* = \arg \max_{\pi} P(x, \pi)$$



# Viterbi Algorithm

- Huge number of state paths, but many subpaths are duplicated so use DP.
- Longest path program in a DAG.

Init ( $i=0$ ):  $v_0(0)=1, v_k(0)=0$  for  $k>0$ .

Step ( $i=1\dots L$ ):  $v_l(i) = e_l(x_i)\max_k(v_k(i-1)a_{kl})$

$\text{ptr}_i(l) = \text{argmax}_k(v_k(i-1)a_{kl})$

Terminate:  $P(x, \pi^*) = \max_k(v_k(L)a_{k0})$

$\pi^*_L = \text{argmax}_k(v_k(L)a_{k0})$

$k, l$  : HMM states

- Most probable state path for complete DNA sequence.

# Forward Algorithm

- What is probability of being in HMM state  $k$  at position  $i$  in a DNA sequence  $x$ ?
- Many different state paths give rise to same sequence  $x$ , so must add the probabilities of all possible paths together.

$$P(x) = \sum_{\pi} P(x, \pi)$$

Init ( $i=0$ ):  $f_0(0)=1, f_k(0)=0$  for  $k>0$ .

Step ( $i=1 \dots L$ ):  $f_l(i) = e_l(x_i) \sum_k f_k(i-1) a_{kl}$

Terminate:  $P(x) = \sum_k f_k(L) a_{k0}$

$$f_k(i) = P(x_1 \dots x_i, \pi_i = k)$$

- Probability of HMM state  $k$  and partial DNA sequence  $x$  up to position  $i$ .
- Need to take into account the probability of remaining sequence.

# Backward Algorithm

- Calculate backward to determine the posterior probability.

$$b_k(i) = P(x_{i+1} \dots x_L \mid \pi_i = k)$$

Init (I=L):  $b_k(L) = a_{k0}$  for all  $k$ .

Step (I=L-1...1):  $b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i+1)$

Terminate:  $P(x) = \sum_l a_{0l} e_l(x_1) b_l(1)$

- Probability of HMM state  $k$  and partial DNA sequence  $x$  from position  $i$  to the end.

# Conclusion

- Probability of being in HMM state  $k$  at position  $i$  for a DNA sequence  $x$ .

$$P(\pi_i = k | x) = \frac{f_k(i)b_k(i)}{P(x)}$$

- Going back to our second question; we can determine the most probable HMM state for each symbol in the DNA sequence. Prediction of whether a particular symbol is or is not part of a CpG island.
- Numeric stability of HMM algorithms
  - Multiplying many of these small probabilities together leads to underflow.
  - Take logarithms of the probabilities so the multiplication turns into summation.
  - Scale the  $f$  and  $b$  variables to keep the value within an acceptable range.
- We have made the assumption throughout all this that we have the transition probabilities, emission probabilities, and HMM structure. Constructing an HMM and estimating parameters is a difficult task.