CSI5126. Algorithms in bioinformatics

Hidden Markov Models (continued)

Marcel Turcotte

uOttawa

School of Electrical Engineering and Computer Science (EECS)
University of Ottawa

Version October 31, 2018

Summary

This module is about **Hidden Markov Models**.

General objective

- **Describe** in your own words Hidden Markov Models.
- **Explain** the **decoding**, **likelihood**, and **parameter estimation** problems.

Reading

- Pavel A. Pevzner and Phillip Compeau (2018) *Bioinformatics Algorithms: An Active Learning Approach*. Active Learning Publishers. `http://bioinformaticsalgorithms.com` Chapter 10.
- Yoon, B.-J. Hidden Markov Models and their Applications in Biological Sequence Analysis. *Curr. Genomics* **10**, 402–415 (2009).
- A. Krogh, R. M. Durbin, and S. Eddy (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press.

HMM: Applications

1. Gene prediction
2. Pairwise and multiple sequence alignments
3. Protein secondary structure
4. ncRNA identification, structural alignments, folding and annotations
5. Modeling transmembrane proteins

Hidden Markov Models (HMM)

"A *hidden Markov model* (HMM) is a statistical model that can be used to describe the evolution of observable events [**symbols**] that depend on internal factors [**states**], which are not directly observable."

"An HMM consists of **two stochastic processes** (...)":

- **Invisible** process consisting of states
- **Visible (observable)** process consisting of symbols

- Yoon, B.-J. Hidden Markov Models and their Applications in Biological Sequence Analysis. *Current Genomics* **10**, 402–415 (2009).

Definitions

We need to distinguish between the **sequence of states** (π) and the **sequence of symbols** (S).
The sequence of states, denoted by π and called the **path**, is modeled as a **Markov chain**, these transitions are not directly observable (they are **hidden**),

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

where a_{kl} is a **transition probability** from the state π_k to π_l.

Each state has **emission** probabilities associated with it:

$$e_k(b) = P(S(i) = b | \pi_i = k)$$

the probability of **observing**/emitting the symbol b when in state k.

Definitions

The **alphabet of emited symbols**, Σ, the **set of (hidden) states**, Q, a **matrix of transition probabilities**, A, as well as a the **emission probabilities**, E, are the parameters of an HMM: $\mathcal{M} =< \Sigma, Q, A, E >$.

Remark

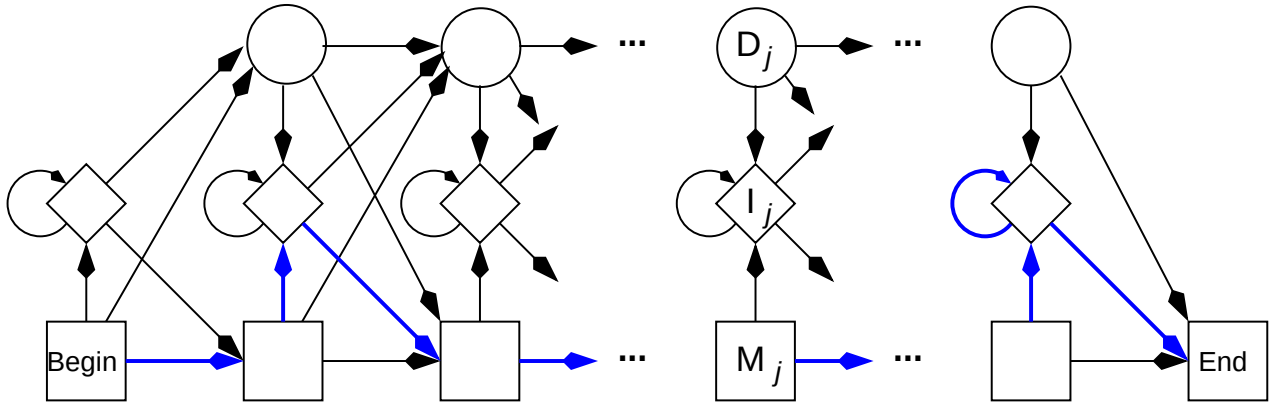
A **path** is modelled as a diescte time-homogeneous first-order Markov chain.

- **Memoryless:** The probability of being in state j at the next time point depends only on the current state, i;
- **Homogeneity of time:** The transition probability does not change over time.

Interesting **questions**

1. $P(S, \pi)$: the joint probability of a sequence of symbols S and a sequence of states π. The **decoding problem** consists of finding a path π such that $P(S, \pi)$ is maximum;
2. $P(S|\theta)$: the probability of a sequence of symbols S given the model θ. It represents the likelihood that sequence S has been produced by this HMM, let's call this the **likelihood problem**;
3. Finally, how are the parameters of the model (HMM), θ, determined? Let's call this the **parameter estimation problem**.

Definitions



Joint probability of a sequence of symbols S and a sequence of states π:

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

$$P(S = \mathrm{VGPGGAHA}, \pi = \mathrm{BEG}, M_1, M_2, I_3, I_3, I_3, M_3, M_4, M_5, \mathrm{END})$$

$\Rightarrow$ However in practice, the state sequence π is not known in advance.

Worked example: the **occasionally dishonest player**

A simplified example will help better understanding the characteristics of HMMs.

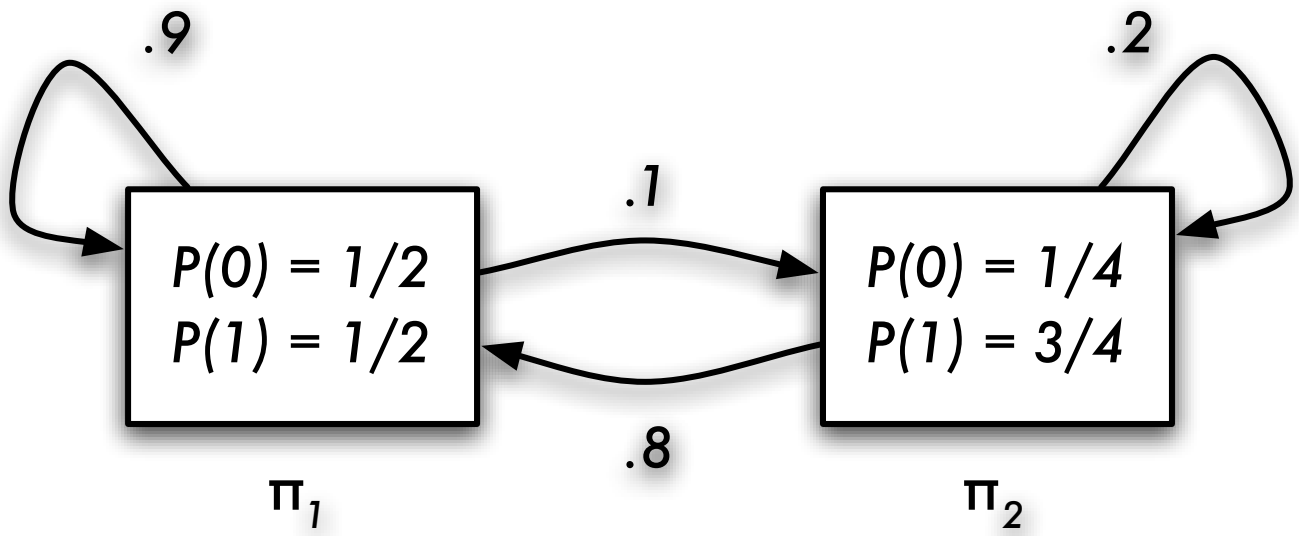
I want to play a game. I will be tossing a coin n times. This information can be represented as follows: { H, T, T, H, T, T, ...} or { 0, 1, 1, 0, 1, 1, ...}.

In fact, I will be using **two coins**! One is **fair**, i.e. head and tail are equiprobable outcomes, but the other one is **loaded** (biased), it returns head with probability $\frac{1}{4}$ and tail with probability $\frac{3}{4}$.

I will not reveal when I am exchanging the coins. This information is hidden to you.

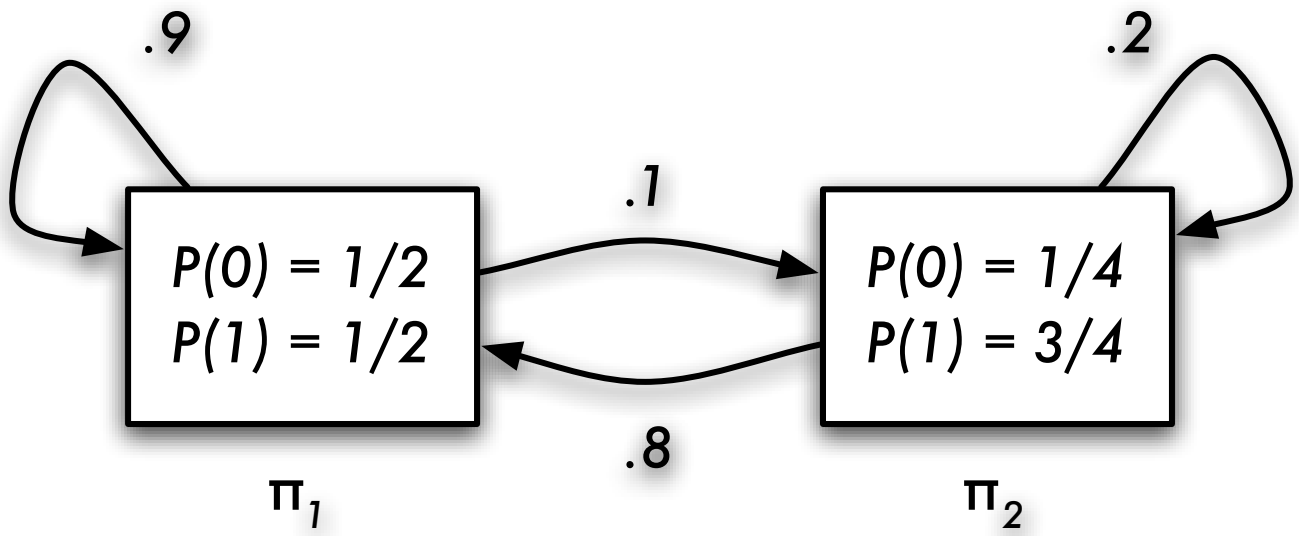
Objective: Looking at a series of observations, S, can you predict when the exchanges of coins occurred?

Worked example: the **occasionally dishonest player**



Such game can be modeled using an HMM where **each state represents a coin**, with its own emission probability distribution, and the transition probabilities represent exchanging the coins.

Worked example: the **occasionally dishonest player**



Given an **input sequence of symbols** (heads and tails), such as 0, 1, 1, 0, 1, 1, 1, which **sequence of states** has the highest probability?

Worked example: the **occasionally dishonest player**

S	0	1	1	0	1	1	1
π	π_1	π_1	π_1	π_1	π_1	π_1	π_1
π	π_1	π_1	π_1	π_1	π_1	π_1	π_2
$\cdots$							
π	π_2	π_2	π_1	π_1	π_2	π_2	π_2
$\cdots$							
π	π_2	π_2	π_2	π_2	π_2	π_2	π_2

Worked example: the **occasionally dishonest player** (cont.)

Since the game consists of printing the series of switches from one coin to the other, selecting the path with the highest joint probability, $P(S, \pi)$, seems appropriate.

Here, there are $2^7 = 128$ possible paths, enumerating all of them is feasible.
However, the number of states and consequently the number of possible paths are generally much larger: $\mathcal{O}(M^L)$, where M is the number of states and L is the length of the sequence of symbols.

The **decoding** problem

Given an observed sequence of symbols, S, **the decoding problem** consists of finding a sequence of states, π, such that the joint probability of S and π is maximum.

$$\text{argmax}_\pi \; P(S, \pi)$$

For our game, the sequence of states is of interest because it serves to predict the exchanges of coins.

The **decoding** problem

If the observed sequence of symbols was of length one, the sequence of states would also be of length one (in our restricted example).

Which state would you predict if the observed symbol was a 0?

What if it was a 1?

Now consider an observed sequence of length two, let's assume that the last symbol is 1, what is the probability of that symbol being emitted from state π_1?

There are two ways of ending up in π_1 while producing $S(2)$: 1) $S(1)$ could have been produced from π_1, and the state remained π_1, or 2) $S(1)$ could have been produced from π_2, and there was a transition π_2 to π_1. The two joint probabilities would be

The **decoding** problem (cont.)

$P(S(1)|\pi_1)P(\pi_1 \to \pi_1)P(S(2)|\pi_1)$ and
$P(S(1)|\pi_2)P(\pi_2 \to \pi_1)P(S(2)|\pi_1)$.

The **decoding** problem

Now consider an observed sequence of length three, let's assume that the last symbol is 1, what is the probability of that symbol being emitted from state π_1?

There are two ways of ending up in π_1 while producing $S(3)$: 1)

the last state that led to the production of the sequence of symbols $S[1,2]$ was π_1 and the state remained π_1, or 2) the last state that led to the production of the sequence of symbols $S[1,2]$ was π_2 and it is followed by a transition π_2 to π_1, with probability a_{21}.

Let's define $v_k(i)$ as **the probability of the most probable path ending in state k while producing the observation i.** Using this notation for formulating the probabilities for the above two scenarios.

$$v_1(3) = \max\left[\, v_1(2) \times a_{11} \times e_1(0),\ v_2(2) \times a_{21} \times e_1(0) \,\right]$$

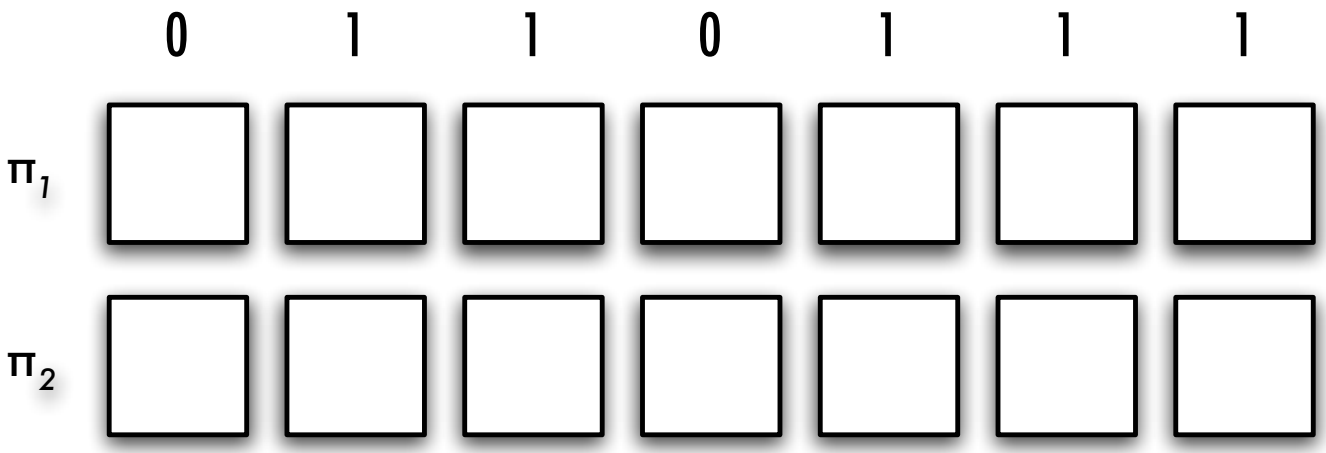
The **decoding** problem

For our 2 states HMM, we can write the following equation,

$$v_1(i) = \max\left[\, v_1(i-1) \times a_{11} \times e_1(S(i)),\ v_2(i-1) \times a_{21} \times e_1(S(i)) \,\right]$$

$$v_2(i) = \max\left[\, v_1(i-1) \times a_{12} \times e_2(S(i)),\ v_2(i-1) \times a_{22} \times e_2(S(i)) \,\right]$$

The **decoding** problem

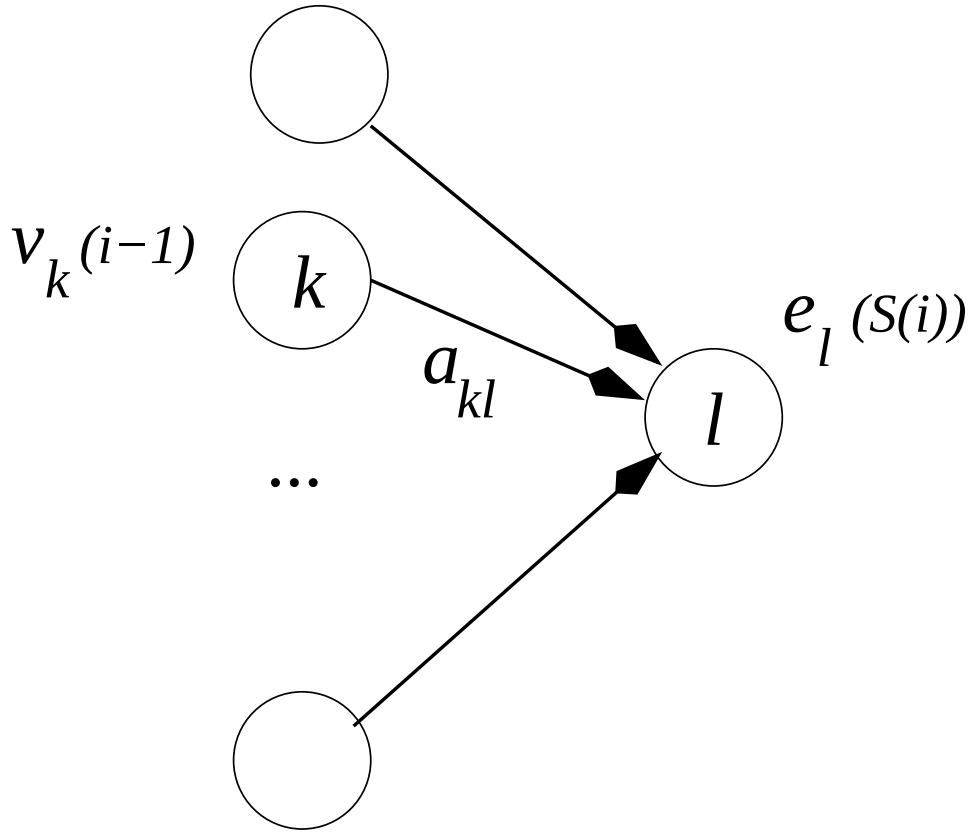


The **decoding** problem

The most probable path can be found recursively. The score for the most probable path ending in state l with observation i, noted $v_l(i)$, is given by,

$$v_l(i) = e_l(S(i)) \max_k [v_k(i-1)a_{kl}]$$

The **decoding** problem (cont.)



where k is running for states such that a_{kl} is defined.

The **decoding** problem

The algorithm for solving the decoding problem is known as the **Viterbi algorithm**. It finds the best (most probable) path using the dynamic programming technique.
Initialization:

$$v_0 = 1, v_k = 0, \ \ k > 0$$

Recurrence:

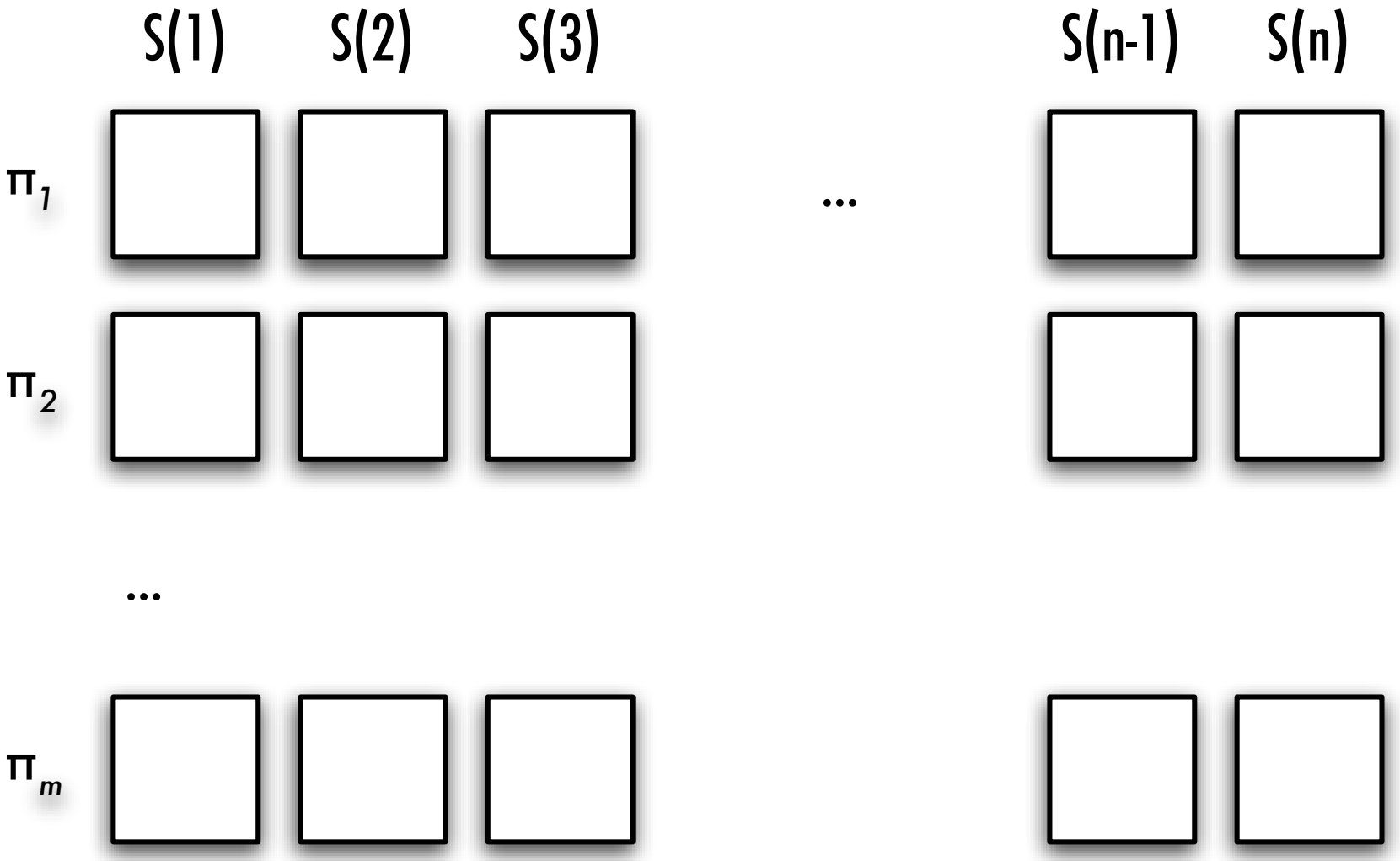
$$v_l(i) = e_l(S(i)) \max_k(v_k(i-1)a_{kl})$$

where, $v_k(i)$ represents the probability of the most probable path ending in state k and position i in S.
A pointer (backward) is kept from l to the value of k that maximizes $v_k(i-1)a_{kl}$.

The **decoding** problem (cont.)

$\Rightarrow$ **Implementation issue:** because of the products (small) probabilities leads to underflow the algorithm is implemented using the logarithm of the values and therefore the products becomes sums.

The **decoding** problem



The **decoding** problem

```
# transition probabilities (t)
$t[0][0] = 0.9; $t[0][1] = 0.1;
$t[1][0] = 0.2; $t[1][1] = 0.8;

# emission probabilities (e)
$e[0][0] = 0.50; $e[0][1] = 0.50;
$e[1][0] = 0.05; $e[1][1] = 0.95;

# observed sequence (S)
@S = (0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1);

# initialization (d is the dynamic programming table)
$d[ 0 ][ 0 ] = $e[ 0 ][ $S[ 0 ] ];
$d[ 1 ][ 0 ] = $e[ 1 ][ $S[ 0 ] ];
```

The **decoding** problem

```
for ( $j=1; $j < @S; $j++ ) {
    for ( $i=0; $i <= 1; $i++ ) {
        $m = 0;
        for ( $k=0; $k <= 1; $k++ ) {
          $v = $d[$k][$j-1]*$t[$k][$i]*$e[$i][$S[$j]];
            if ( $v > $m ) {
                $from = $k; $to = $i; $m = $v;
            }
        }
        $d[ $i ][ $j ] = $m;
        $tr[ $i ][ $j ] = "($from->$to)";
    }
}
```

The **decoding** problem

```
for ( $i=0; $i <= 1; $i++ ) {
    for ( $j=0; $j < @S; $j++ ) { printf "\t%5.5f", $d[ $i ][ $j ]; }
    print "\n";
    for ( $j=0; $j < @S; $j++ ) { printf "\t %s", $tr[ $i ][ $j ]; }
    print "\n";
}
```

The **decoding** problem

t[0][0] = 0.9; t[0][1] = 0.1; t[1][0] = 0.2; t[1][1] = 0.8;
e[0][0] = 0.50; e[0][1] = 0.50; e[1][0] = 0.05; e[1][1] = 0.95;

0	1	0	1	0	1	1	1	1	1	1	1
0.50000	0.22500	0.10125	0.04556	0.02050	0.00923	0.00415	0.00187	0.00084	0.00038	0.00017	0.00008
	(0->0)	(0->0)	(0->0)	(0->0)	(0->0)	(0->0)	(0->0)	(0->0)	(0->0)	(0->0)	(0->0)
0.05000	0.04750	0.00190	0.00962	0.00038	0.00195	0.00148	0.00113	0.00086	0.00065	0.00049	0.00038
	(0->1)	(1->1)	(0->1)	(1->1)	(0->1)	(1->1)	(1->1)	(1->1)	(1->1)	(1->1)	(1->1)

The **decoding** problem

- Given an HMM representing a protein family as well as an unknown protein sequence, the solution to the decoding problem reveals the internal structure of the unknown sequence, showing the location of the insertions and deletions, core elements, etc.;

The **likelihood** problem: calculating $P(S|\theta)$

In the case of a Markov chain there is a single path for a given sequence S and therefore $P(S|\theta)$ is given by,

$$P(S|\theta) = P(S(1)) \prod_{i=2}^{n} a_{S(i-1)S(i)}$$

In the case of an HMM, there are several paths producing the same S (some paths will be more likely than others) and $P(S|\theta)$ should be defined as the sum of all the probabilities of all possible paths producing S,

$$P(S|\theta) = \sum_{\pi} P(S, \pi)$$

The number of paths grows exponentially with respect to the length of the sequence, therefore all the paths cannot simply be enumerated and summed.

The **likelihood** problem: forward algorithm

Modifying the Viterbi algorithm changing the maximization by a sum calculates the probability of the observed sequence up to position i ending in state l,

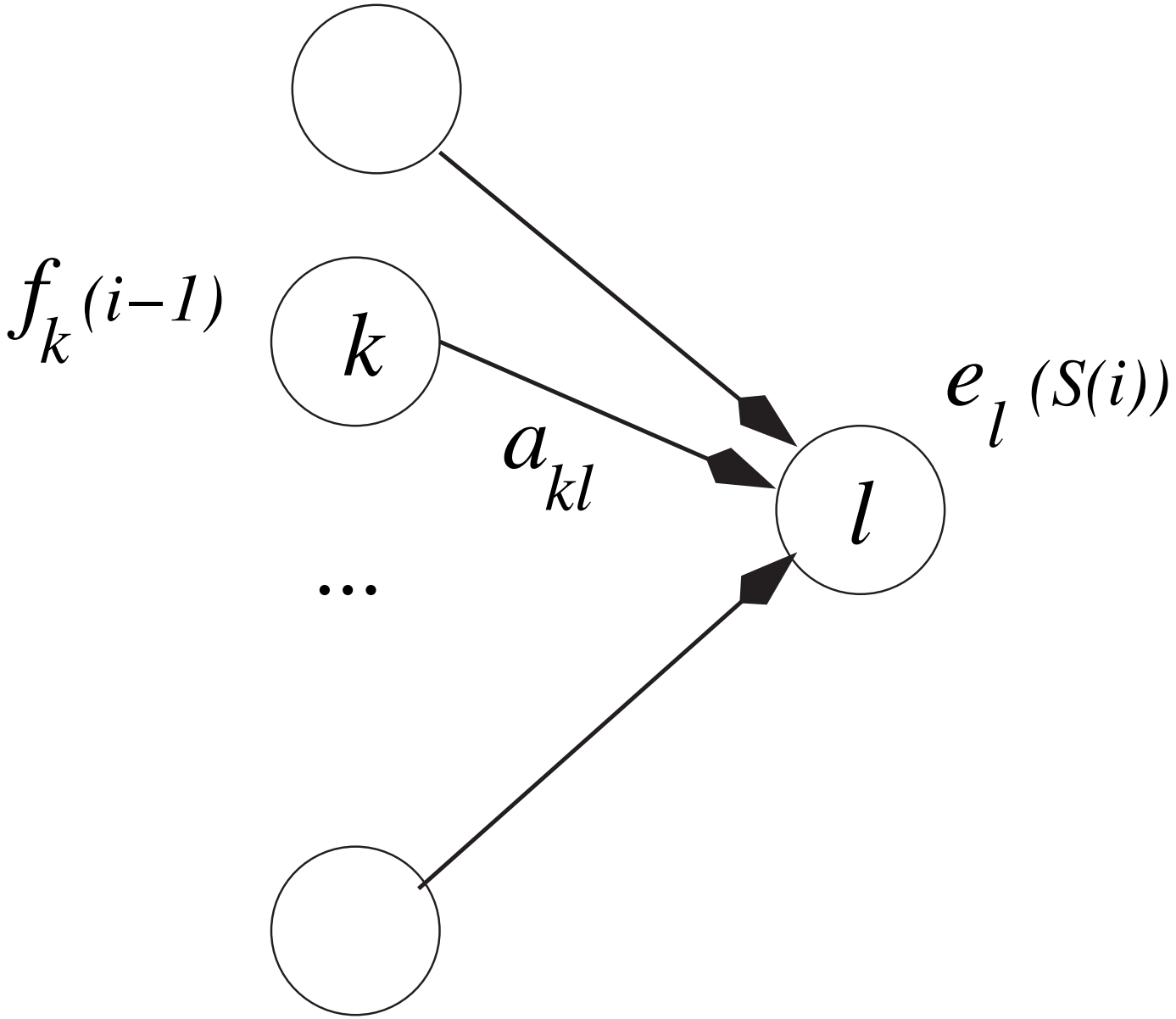
$$f_l(i) = e_l(S(i)) \sum_k f_k(i-1) a_{kl}$$

The **likelihood** problem

The score represents the probability of the sequence up to (and including) $S(i)$, noted $f_l(i)$, is given by,

$$f_l(i) = e_l(S(i)) \sum_k [f_k(i-1)a_{kl}]$$

The **likelihood** problem (cont.)



Forward Algorithm

Can you think of an application for the **forward algorithm**?

Pfam is a large collection of HMMs covering many common protein domains and families, one HMM per domain or family, version 30.0 (June 2016) contains 16306 families.

Given a new sequence, the forward algorithm can be used for finding the family that it belongs (if any).

$\Rightarrow$ pfam.xfam.org

Model Specification

We now turn to our third and final question. How to determine the parameters of the model?

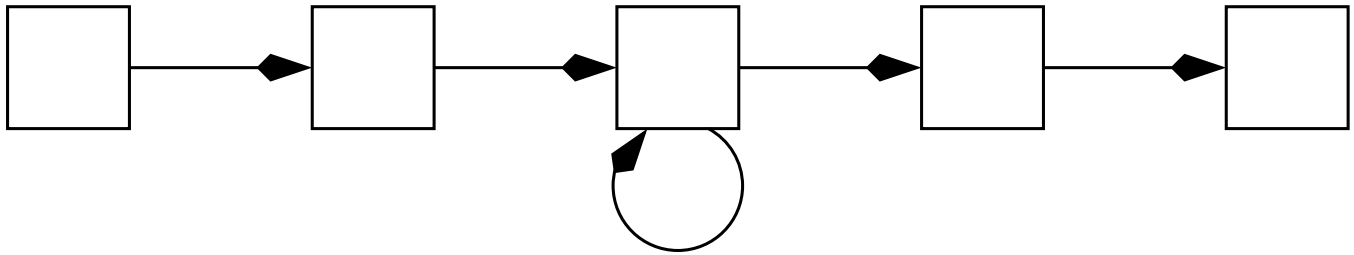
Let $x_1, \ldots, x_m$ be m independent examples forming the training set (typically, m sequences), the objective is to find a set parameters, θ, such that

$$\max_{\theta} \Pi_{i=1}^{m} P(x_i|\theta)$$

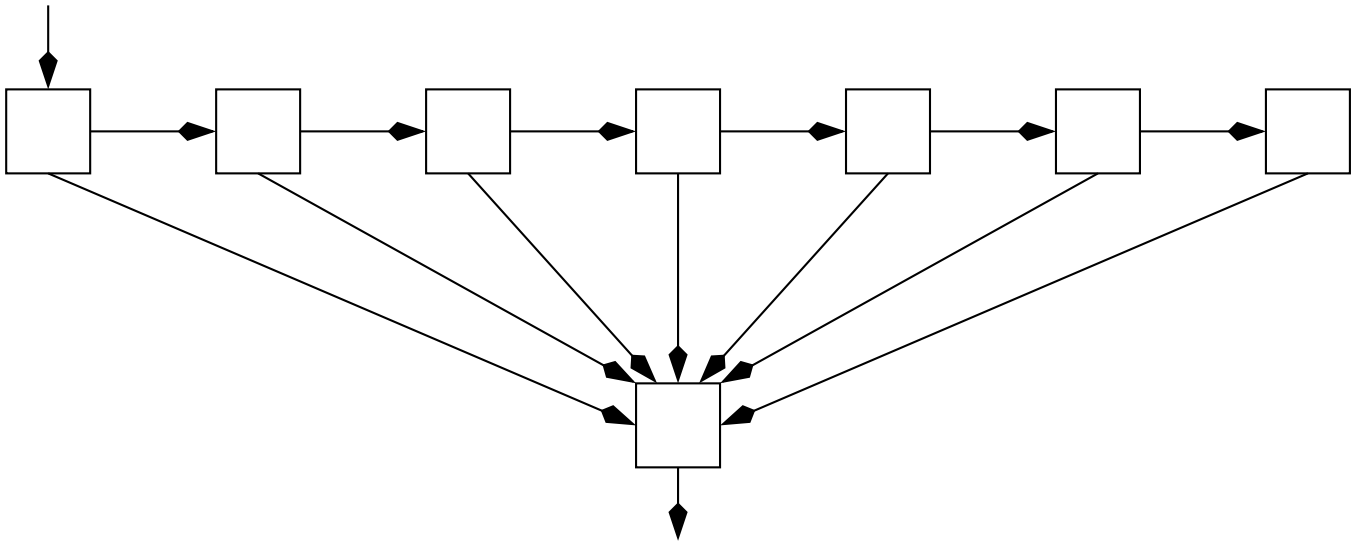
Model Specification

- Structure: states + interconnect;
  (This is an occasion to include domain specific information!)
- Estimating the transition/emission probabilities.

Modeling the **length**

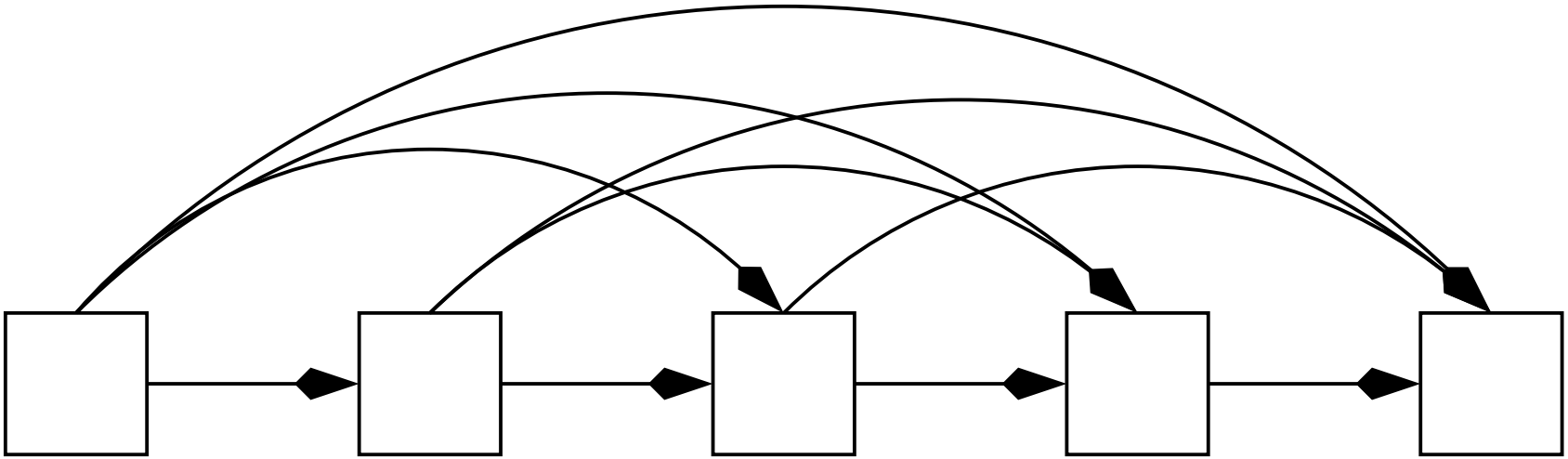


At least 5 symbols long



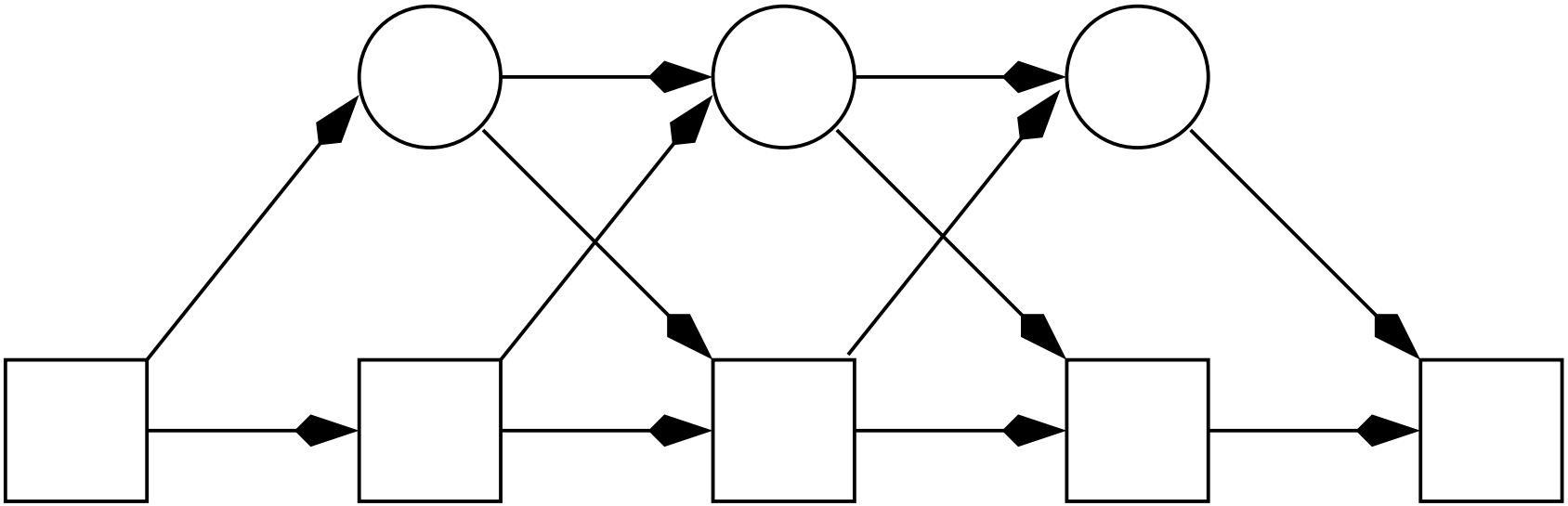
2 to 8 symbols long

Arbitrary **Deletions**



Too expensive, too many parameters to evaluate!

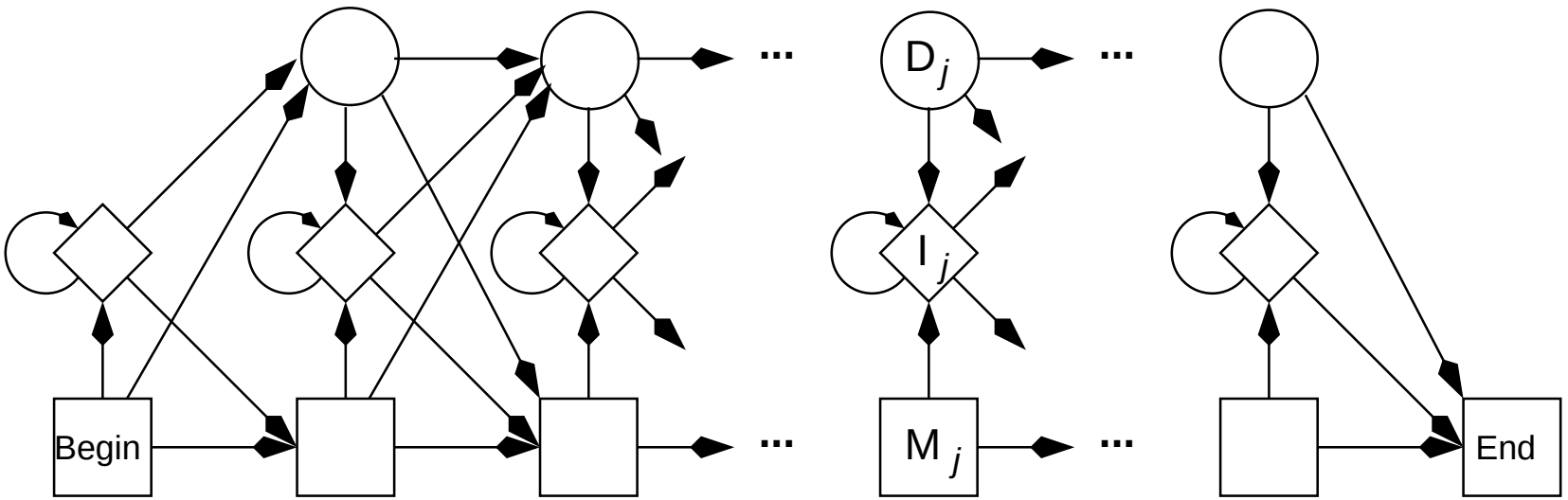
Arbitrary **Deletions** (cont.)



Silent (null) states do not emit symbols.

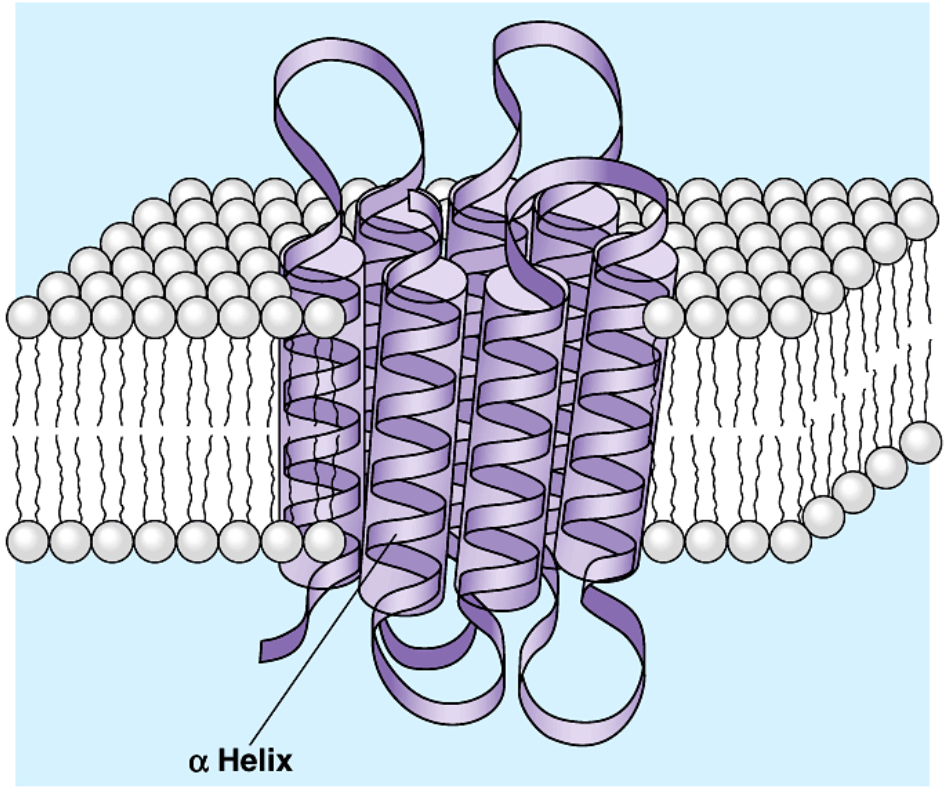
$\Rightarrow$ Silent states prevent modeling specific distant transitions.

Profile HMMs



$\Rightarrow$ Models insertion/deletions separately.

Trans-membrane (helical) proteins



Copyright © Pearson Education, Inc., publishing as Benjamin Cummings.

Trans-membrane (helical) proteins (cont.)

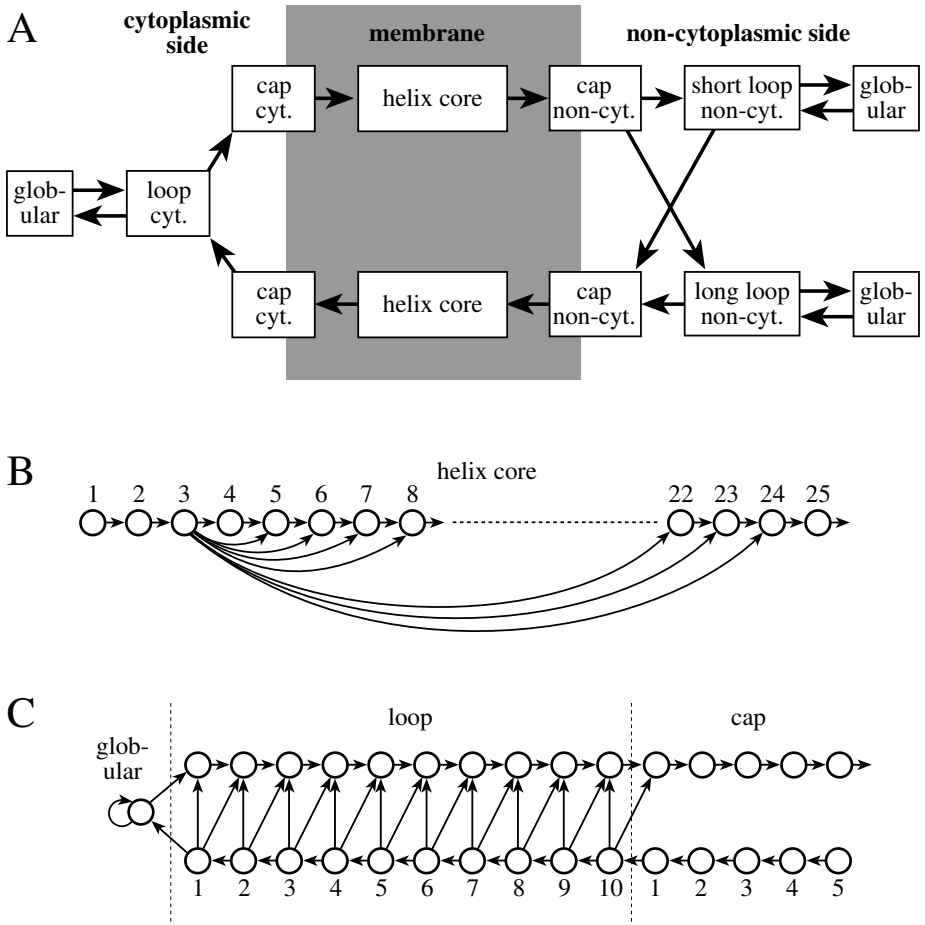


Figure 1: The structure of the model used in TMHMM. A) The overall layout of the model. Each box corresponds to one or more states. Parts of the model with the same text are tied, i.e. their parameters are the same. Cyt. means the cytoplasmic side of the membrane and non-cyt. the other side. B) The state diagram for the parts of the model denoted helix core in A. From the last cap state there is a transition to core state number 1. The first three and the last two core states have to be traversed, but all the other core states can be bypassed. This models core regions of lengths from 5 to 25 residues. All core states have tied amino acid probabilities. C) The state structure of globular, loop, and cap regions. In each of the three regions the amino acid probabilities are tied. The three different loop regions are all modelled like this, but they have different parameters in some regions.

Gene prediction

Gene prediction (cont.)

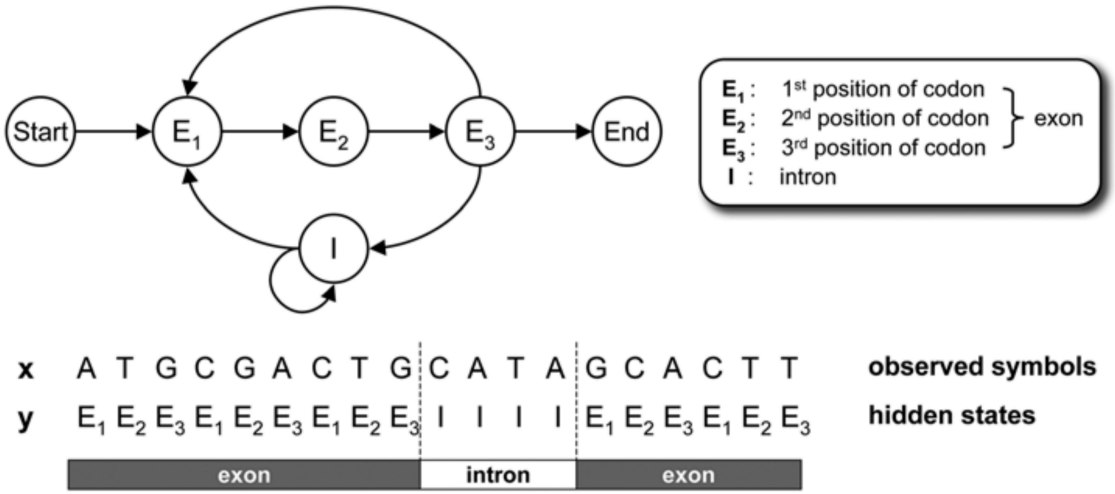
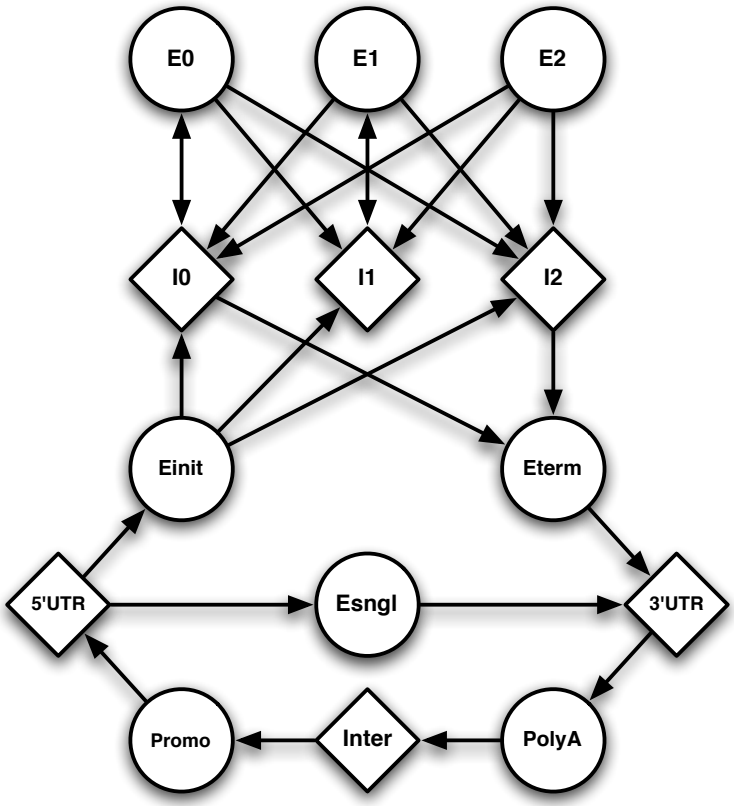


Fig. (1). A simple HMM for modeling eukaryotic genes.

Gene prediction (cont.)



$\Rightarrow$ genes.mit.edu/GENSCAN.html

The **parameter estimation** problem

Problem: estimate the a_{st} and $e_k(b)$ probabilities.
Given:

- a fixed topology;
- n independent positive examples: $S_1, S_2, \ldots, S_n$.

$$\log P(S_1, S_2, \ldots, S_n|\theta) = \sum_{j=1}^{n} \log P(S_j|\theta)$$

Two scenarios:

- The paths are known
  (CG islands, secondary structure, gene prediction);
- The paths are unknown.

Parameters **estimation**/known paths

- Maximum likelihood estimators are

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \text{ and } \frac{E_k(b)}{\sum_{b'} E_k(b')}$$

  - Necessitates large number of positive examples;
  - If a state k is not visited than numerator and denominator are zero;
  - $P(x, \pi)$ is a product of probabilities, what happen if an arc/emission is zero?

- Work around?
  - $A_{kl} = A_{kl} + r_{kl}$
  - $E_k(b) = E_k(b) + r_k(b)$

Parameters **estimation**/known paths (cont.)

where r_{kl} and $r_k(b)$ are pseudocounts. The simplest pseudo count would be $r_{kl} = 1$ and $r_k(b) = 1$. Better pseudocounts would reflect our prior bias, using observed frequency of amino acids or derived from substitution scores.

Parameters **estimation**: remarks

Some (emission, transition) probabilities can be zero, if this is the case then all path involving those probabilities would have probability zero as well. In particular, this would happen if the number of sequences used to build the model is low, "strong conclusions would be drawn from very little evidence".
To circumvent that problem, pseudocounts are added prior to calculating the frequencies. The simplest pseudocounts consist in initializing all the counts to one; rather than zeros before counting the number of occurrences of each event.
In the case of the emission probabilities, this would be assuming that all amino acids are equiprobable. Since counts don't need to be integers, a solution would be to initialize the counts with a value between zero and one, proportional to the overall distribution of the amino acids.

Parameters **estimation**: remarks

More sophisticated pseudocounts would reflect the distribution of the amino acids at that position. For example, if leucine occurs with a high frequency at that position, you would expect that isoleucine would occur with a high frequency too, but not arginine — in the PAM250 scoring matrix, the score for substituting leucine and isoleucine is 2.80 whilst the score for leucine arginine is -2.2.

Parameters **estimation**/unknown paths

It is also possible to estimate the emission/transition probabilities when the paths are unknown.

In the case of profile-HMMs, it is possible to estimate the parameters of the HMM starting with a set of unaligned sequence.

The details of these methods are complex, but the general idea is as follows: the model is initialized with more or less random values (we say more or less because one can use prior knowledge about the distribution of the amino acids or a rough sequence alignment as a starting point).

The model is used to aligned the sequences from the training set, the alignment is then used to improve the parameters of the model. The "improved" model is used to align the training sequences again, in general, this will lead to a slightly improved alignment, which is used again to improve the probabilities of the

Parameters **estimation**/unknown paths (cont.)

model, the process is repeated until no improvement of sequence alignment is observed.
The scheme for parameter estimation is called "Expectation-Maximization", one of the standard algorithms for model estimation is called Baum-Welch or forward-backward algorithm.
One the main problem or limitation with this technique is that it converges toward a local optimum, i.e. it is not guaranteed to find the most probable model given the observed data.

Expectation-Maximization (EM) algorithm

1. Choose an initial model. If no prior information is available, make all the transition probabilities equiprobable, similarly for the emission probabilities;
2. Use the decoding algorithm for finding the maximum likelihood path for each input sequence;
3. Using these alignments, tally statistics for estimating all a_{kl} and $e_k(b)$ values;
4. Repeat 3 and 4 until the parameter estimates converge.

Summary

Like Markov Chains, Hidden Markov Models (HMMs) consist of a finite number of states, $\pi_1, \pi_2, \ldots$, and transition probabilities, $P(\pi_i \rightarrow \pi_j)$.

Unlike Markov Chains, HMMs also "emit" a symbol (letter) at each (most) states.

Sequence of states $\pi = \pi_1, \pi_2 \ldots$
Sequence of observed symbols $S = S(1), S(2) \ldots$
Given a new observation, the sequence of symbols is known (observed) but not the sequence of states, "it is hidden".

Historical note

HMMs were first developed for solving speech recognition problems in the early 1970s.

1. A speech signal is divided into frames of 10 to 20 milliseconds;
2. A process called vector quantization assigns a predefined category to each frame (typically 256 predefined categories);

The input is now represented as a long sequence of category labels (symbols).
The next task is to recognize words in this long sequence of categories.
However, variations are observed (that will be seen as category substitutions, insertions and deletions).
HMMs were developed in such context.

Historical note (cont.)

Example: CG islands

[From Durbin *et al.* Biological Sequence Analysis.]
Certain regions of the human genome are known as CG (or CpG) islands.
These regions, located around promoters or start regions of many genes, show a higher frequency of CG dinucleotides than elsewhere*.
Those regions are a few hundred to a few thousand bases long.

*this is because whenever C is followed by G the chances that C will be methylated are higher (adding CH_3 group to its base), also, methylated Cs mutate to T with high frequency, therefore CG dinucleotides are observed less frequently than expected by chance, $P(C) \times P(G)$, finally, methylation is suppressed in biologically important regions such as the start of a gene which explains the fact the CGs occurs there more frequently then elsewhere.

Example: CG islands

Problem 1: Given an unlabeled (short) sequence of DNA, can we decide if it comes from a CG island or not?

$\Rightarrow$ promoter: a site on DNA to which RNA polymerase will bind and initiate transcription.

Example: CG islands (cont.)

```
LOCUS       AL162458   150791 bp    DNA             PRI       29-SEP-2000
DEFINITION  Human DNA sequence from clone RP11-465L10 on chromosome 20.

     misc_feature    20670..21997
                     /note="CpG island"

    20641                                ...C GCGCGGGTGC CAGGACCCAG GTCCTTGCTA
    20701 CGTCCGGAGC CTACGTCACC ACGATGCCTC CCCTGGGCCG GCGGCAGAAC CCGAGACCCC
    20761 CGCAGGTTCT AAGACAGCCC CCACGCCCCC CAGTGCGCAC GCTCAGTCCA ACCCCGCCGC
    20821 GCACCGCCCA CCGCGAACAT CCGGCTCCTG CGTGTGTGCT CGAGGGGGAA ACTGAGGCGG
    20881 GGACGTGCCA GTGAATTCAT TCCTTCCTCA GTCCACCCGC AGGCCTACAA AGCTGTCTCC
    20941 CCTTCCTCAG CGCCACAAGG AACAGCAGGG ACGGATGGGA AGAAGGGGAG GGGGCCGAAA
    21001 GCAAGCTGGG TGCGAGGAGC CAGCCGACCC TGCCACACTC AAGATGGCGG CGCGGCCGCG
    21061 GCGAGGTCCC TCAGAGGCGG TACCAGCGCA TGCGCAGCGC GGAGTCCCGG CCCGGGACAC
    21121 AAGATGGCGG CAGCGGCGCT GGGGAGGGCG AGGCGGAGGC GGCAAAACGG GCGGTCGAGC
    21181 AGAACGTGTA GCCGCGTCCC CTCCAGTCCG CTCCGGGCAG GTAAGAGTCC CAGGAAGCCA
    21241 TGGTCCCGCA GCGAGCCGCG CCAGGGTCTG GGGATCCGAA GCTGGGGGGC GGCGGCCCCT
    21301 CCGGCGCTTT CTGCTCGGGA CTGCCGCTTG CCCTGTCTCT GTTGCCGCCG CCATCTTAGA
    21361 CCCGCGGGTG GGCGGCCGCG CCGGTGGCCG AAGTGAGGGA GGTGGGCCCG GAGAGCCCCA
    21421 GCGGAGCGGG CTCTAGGGCC CCTCCGCTGC TGCCGCCGCC ACCGCCTTTG TGTCGGGCTC
    21481 CGACTCTGAG TCGCCTCAGC CCGGGGGCGG GAGCGCGCGG CGGGGCGGGG GGCGGAGCCC
    21541 GAGAGATGGG CCGGCGCGCG CGCGCGCGCC AAACAGCCCA CCCTCGCTGG GGTAGGGGGA
    21601 GGGGAAGGTG CGCGCGCGCG CGCGCGCTGG AGCTCGCCTC TCGCCTTCGT GCGCCGTCGC
    21661 GCCTGCGTAC TTTGTTCGCC CTTTGACTCC TCCCTACTGG GCCGGAGAAT TCTGATTGGT
    21721 ACATTGCGGA GATGGTCCCG CCCCACGTGC CTCCAATCCC GGACTCGGAC TCTGGCTTCT
    21781 GGTGGGTTTT TCTGGTTGCG CAGATAGAGT TGTTTATCCT TGAGCAGCGG TAATTCTCAA
    21841 ACTGCGGTAT GCGTGGGGGT CGGGAAGCCA CAGGATAAAT AAAGACGTTA ACTTAAGAGC
    21901 AGTTATGTCT TACTGGGAGC GTACAATGCT GGACTCTACA TATAACGGTC GAGTGATTCC
    21961 GGTTTATAAG CCGGAAAGCA GAAGGGCCCG GAATCCG...
```

Probabilistic model of a sequence

$$\begin{aligned} P(x) &= P(x_L, x_{L-1}, \ldots, x_1) \\ &= P(x_L | x_{L-1}, \ldots, x_1) P(x_{L-1} | x_{L-2} \ldots, x_1) \ldots P(x_1) \end{aligned}$$

(by application of the general multiplication rule)

For example the probability of CGAT:

$$\begin{aligned} P(CGAT) &= P(T, A, G, C) \\ &= P(T | A, G, C) P(A | G, C) P(G | C) P(C) \end{aligned}$$

Probabilistic model of a sequence

Under the assumption that positions are independent from one another,

$P(x_i|x_{i-1}\ldots,x_1) = P(x_i)$,

$$\begin{aligned} P(x) &= P(x_L|x_{L-1},\ldots,x_1)P(x_{L-1}|x_{L-2}\ldots,x_1)\ldots P(x_1) \\ &= P(x_L)P(x_{L-1})\ldots P(x_1) \end{aligned}$$

$$\begin{aligned} P(CGAT) &= P(T|A,G,C)P(A|G,C)P(G|C)P(C) \\ &= P(T)P(A)P(G)P(C) \end{aligned}$$

Markov Chains

However under the assumption of an underlying (first order) Markovian process (memory less), $P(x_i|x_{i-1}\ldots,x_1) = P(x_i|x_{i-1})$, and the previous equation can be rewritten as follows:

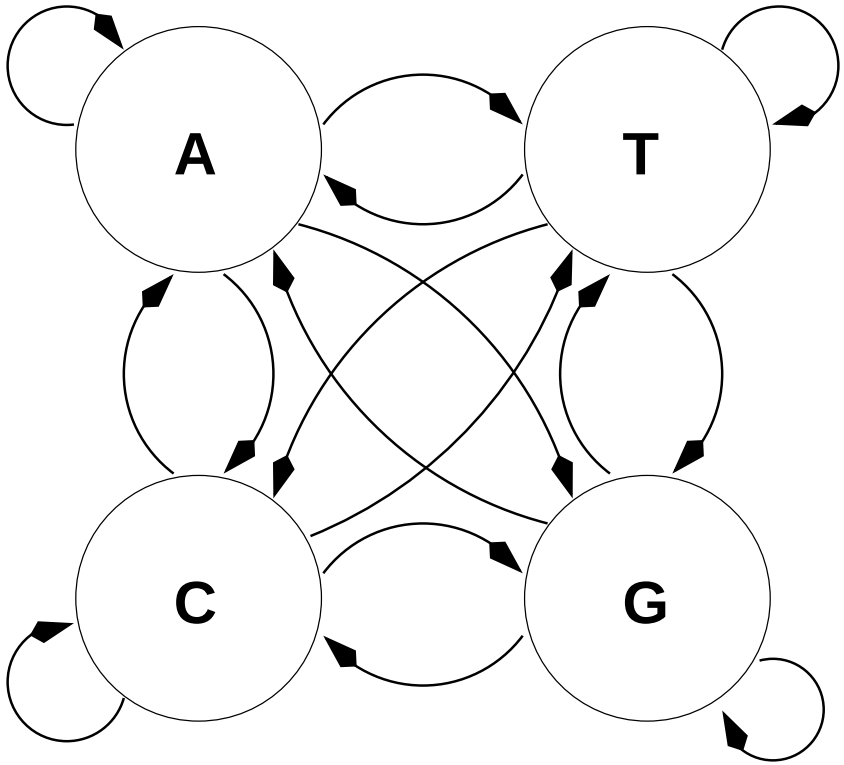
$$\begin{aligned} P(x) &= P(x_L|x_{L-1},\ldots,x_1)P(x_{L-1}|x_{L-2}\ldots,x_1)\ldots P(x_1) \\ &= P(x_L|x_{L-1})P(x_{L-1}|x_{L-2})\ldots P(x_2|x_1)P(x_1) \end{aligned}$$

In the previous example:

$$\begin{aligned} P(CGAT) &= P(T,A,G,C) \\ &= P(T|A)P(A|G)P(G|C)P(C) \end{aligned}$$

Graphical Formalism for Markov Chains

Graphical Formalism for Markov Chains (cont.)



Graphical Formalism for Markov Chains (cont.)

$$a_{st} = P(S(i) = t | S(i-1) = s).$$

$\Rightarrow$ *transition probabilities*, a_{st}, are associated with the arcs of this graph.

Markov Chains

$a_{st} = P(S(i) = t|S(i-1) = s)$ is the probability that symbol t is observed at position i knowing that s occurs at position $i-1$. Therefore $P(S)$ can now be written as follows,

$$P(S) = P(S(1)) \prod_{i=2}^{n} a_{S(i-1)S(i)}$$

Here the concept of time (involved in the development of PAM matrices) has been replaced by that of space, with similar observations,

- *memory less*: the probability that symbol a occurs at position i depends only on what symbol is found at position $i-1$; and not any other $i' < i-1$.
- *homogeneity of space*: the probability that symbol a occurs at position i does not depend on the particular value of i (e.g. $i = 123$ or $i = 162, 144$).

Markov Chains (contd)

Higher-order Markov models are interesting for modeling DNA sequences; in particular for modeling coding regions, the codon structure.

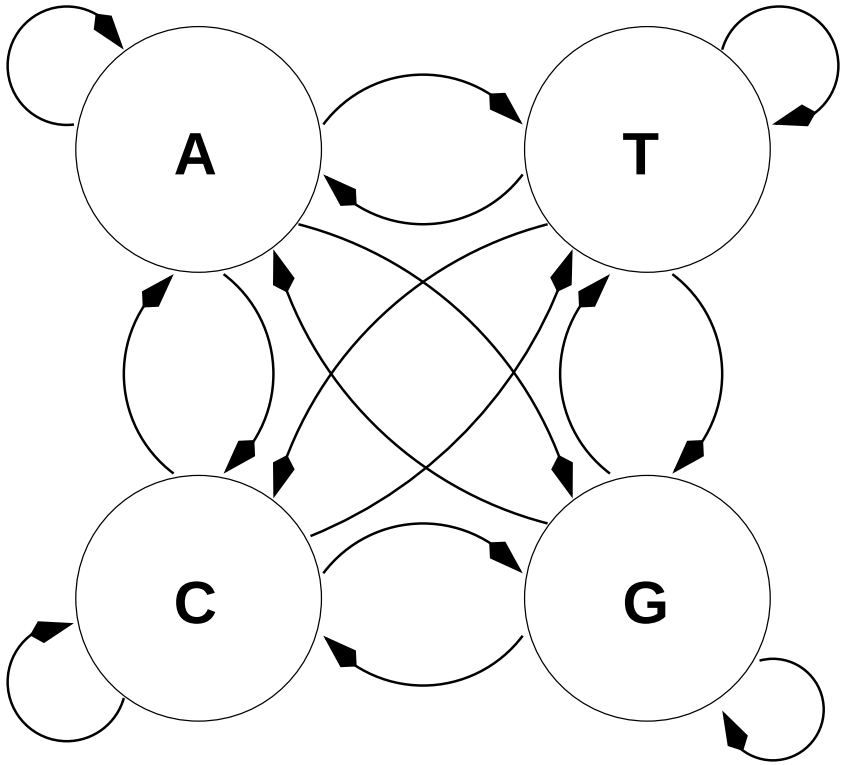
A Markov chain of order k is a model where the probability that symbol a occurs at position i depends only on what symbol is found at positions $i-1, i-2 \ldots i-k$, and not any other $i' < i-k$.

Markov Chains (contd)

Markov chains are particularly convenient for two reasons,

- $P(S(i)|S(i-1)\dots S(1))$ would be difficult to estimate (do you see why?);
- They lead to computationally efficient algorithms.

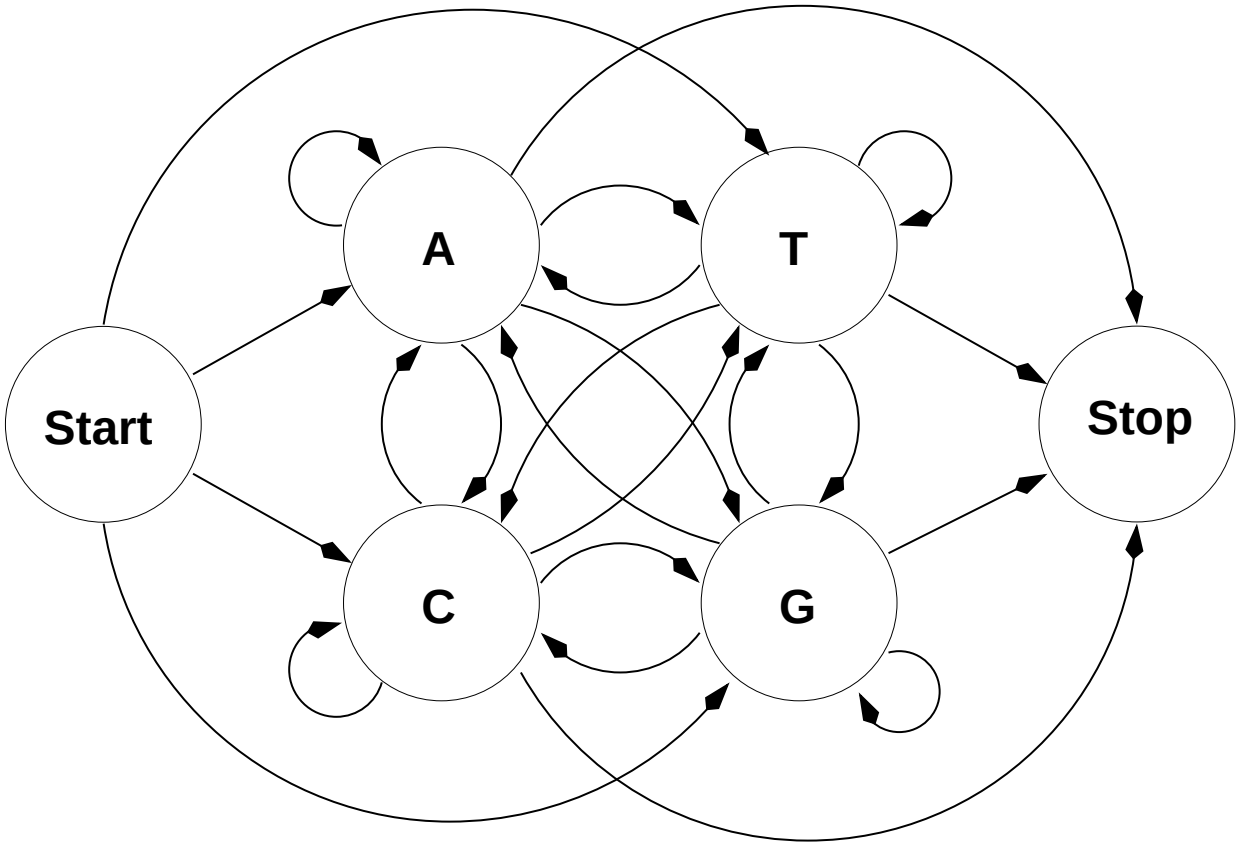
Markov Chains (contd) (cont.)



Markov Chains (contd) (cont.)

$\Rightarrow$ In the above model a sequence can start and end anywhere.

Markov Chains (contd) (cont.)



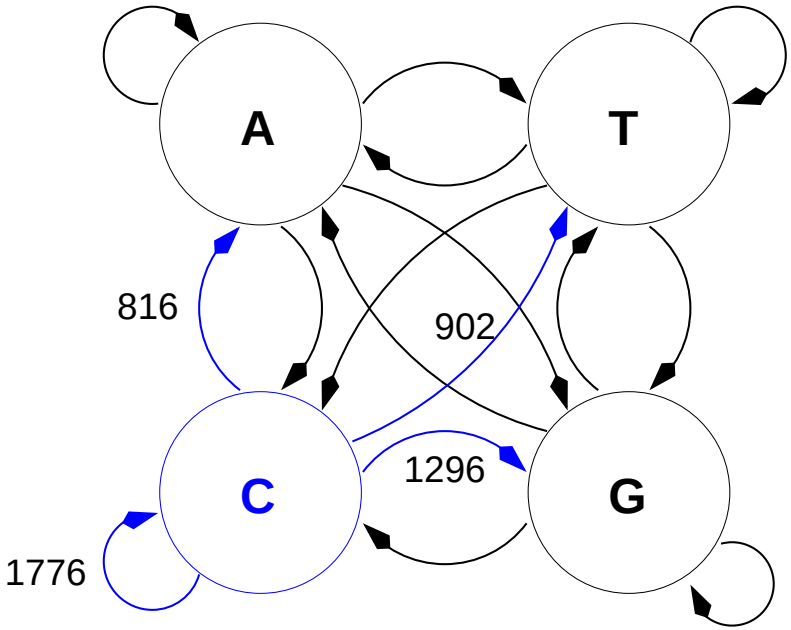
Markov Chains (contd) (cont.)

$\Rightarrow$ 1) Allows modeling start/end effects, $P(Stop|T)$ could be different than $P(Stop|G)$, 2) models the distribution of lengths of the sequences, 3) defines a probability distribution of all possible sequence (of any length)(sum to 1), 4) lengths decays exponentially.

Methodology

- Durbin et al. collected a large number of positive and negative examples of CG islands, almost 60,000 nucleotides in all;
- Construct a Markov Model for the positive examples and one for the negative examples, this involves estimating the transition probabilities;
- To use the models for discrimination, calculate the log-odds ratio.

Maximum Likelihood Estimators



$$a_{st}^{+} = \frac{c_{st}^{+}}{\sum_{t'} c_{st'}^{+}}$$

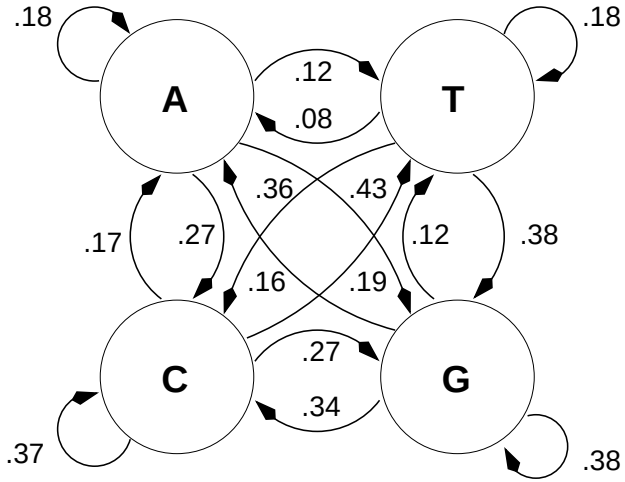
$$\Rightarrow a_{\mathrm{CA}}^{+} = 816/(816 + 902 + 1296 + 1776) = 0.17$$

Maximum Likelihood Estimators

+	A	C	G	T
A	0.180	0.274	0.426	0.120
C	0.171	0.368	0.274	0.188
G	0.161	0.339	0.375	0.125
T	0.079	0.355	0.384	0.182

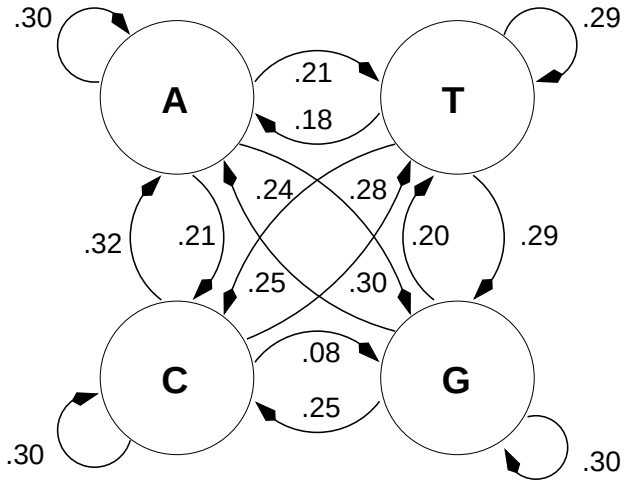
−	A	C	G	T
A	0.300	0.205	0.285	0.210
C	0.322	0.298	0.078	0.302
G	0.248	0.246	0.298	0.208
T	0.177	0.239	0.292	0.292

Maximum Likelihood Estimators (cont.)



$\Rightarrow$ Markov Model for the positive examples of CG islands.

Maximum Likelihood Estimators (cont.)



$\Rightarrow$ Markov Model for the negative examples of CG islands.

Discrimination

To test if a sequence S of length n is likely to be a CG island, the log-odds ratio of the two models is computed,

$$\log \frac{P(S|Model+)}{P(S|Model-)} = \sum_{i=1}^{n} \log \frac{a^{+}_{S(i-1)S(i)}}{a^{-}_{S(i-1)S(i)}}$$

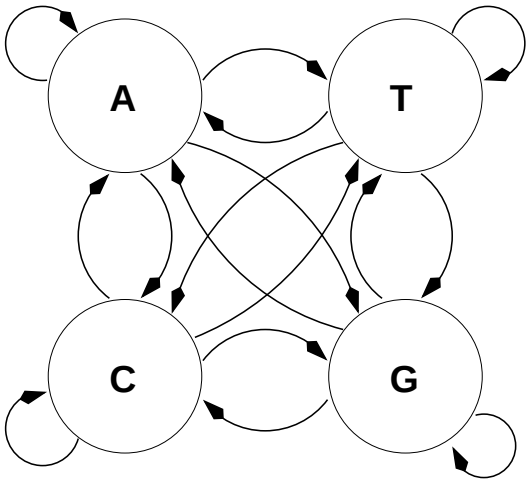
which could also be written as,

$$\sum_{i=1}^{n} \log s(S(i), S(i-1))$$

where,

$$s(s,t) = \frac{a^{+}_{st}}{a^{-}_{st}}$$

Summary

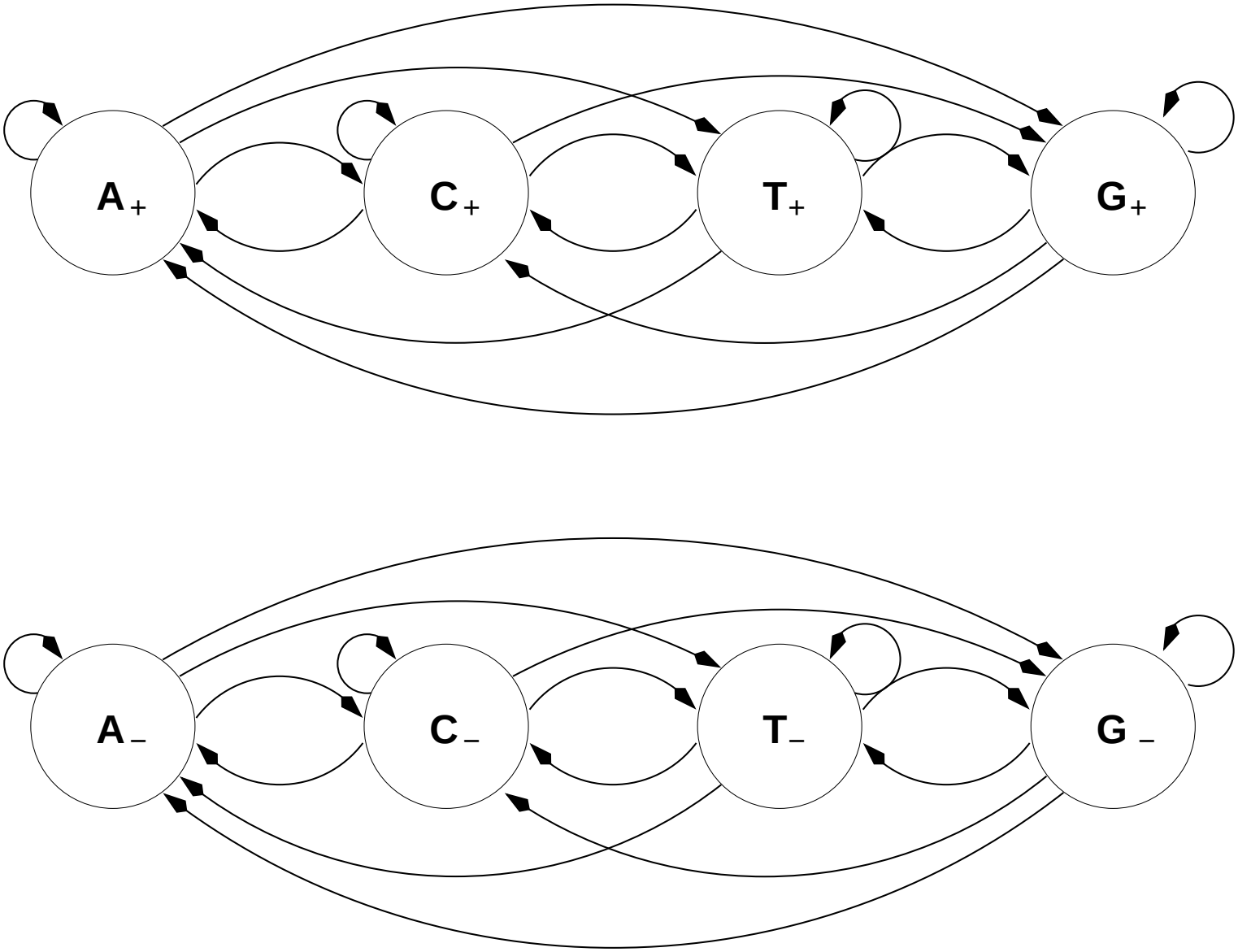


- Each state is associated with a single symbol;
- Models dependencies between adjacent positions;
- Transition probabilities a_{CT}.

Pitfalls

- MM for CG islands can only test entire sequences;

Pitfalls (cont.)



Ideas

- In this case, small probabilities of switching from one model to the other;

CG island (contd)

In the case of the hidden Markov model for the CG island the probabilities of emission are all 1 or 0.
The probability that a sequence CGCG being emitted by the following path in our model C_+, G_-, C_-, G_+ is given by,

$$a_{\text{start},C_+} \times 1 \times a_{C+,G_-} \times 1 \times a_{G_-,C_-} \times 1 \times a_{C_-,G_+} \times 1 \times a_{G_+,\text{end}}$$

CG island (contd) (cont.)

In fact, finding the path π such $P(S, \pi)$ is maximum is often the goal.
What does it mean?
Assuming the transition and emission probabilities are known for the CG island HMM.
Given a new/unlabeled sequence S, i.e.

$$S = CGCCG \ \ldots \ CGCATG$$

we don't know which state was used to emit a given symbol, for example was the first C emitted from C_+ or C_-, was the G second position emitted from G_+ or G_-, and so on, find the path π such that $P(S, \pi)$ is maximum will tell us what are the most likely locations for the CG islands:

$$S = C_+ G_+ C_+ C_+ G_+ \ \ldots \ C_+ G_+ C_- A_- T_- G_+$$

CG island (contd) (cont.)

.

The most probable path:

$$\pi^\star = \text{argmax}_\pi P(S, \pi)$$

Viterbi: Dynamic Programming Table

		C	G	C	G
Start	1	0	0	0	0
A_+	0	0	0	0	0
C_+	0	0.13	0	0.012	0
G_+	0	0	0.034	0	0.032
T_+	0	0	0	0	0
A_-	0	0	0	0	0
C_-	0	0.13	0	0.0026	0
G_-	0	0	0.010	0	0.00021
T_-	0	0	0	0	0

Viterbi

The "best path", sequence of states, determined by the Viterbi algorithm identies the CG islands within a genomic sequence.

Viterbi (cont.)

Probabilistic Models

- Hypothesis: positions are independent from one another
- Pairwise alignment (PAM, BLOSUM, etc.)
  Given two aligned sequences, S'_1 and S'_2, the score of an alignment is given,

$$\sum_{i=1}^{n} s(S'_1(i), S'_2(i))$$

- Position specific scoring scheme
  Given a sequence S and a probabilistic model M of a sequence family/motif, represented a matrix, f, of size $20 \times n$, where $f(a, i)$ represents the probability that amino acid a occurs at position i in this family/motif,

$$P(S|M) = \prod_{i=1}^{n} f(S(i), i)$$

Probabilistic Models (cont.)

or a log-odds score: $\sum_{i=1}^{n} s(S(i), i)$, where $s(a, i) = \log \frac{f(a,i)}{q_a}$.

- Markovian models: Markov chains and hidden Markov models
  - Modelling the distribution of the amino acids within gaps and their length.

Pairwise (Multiple) Alignment vs Profiles vs Hidden Markov Models

- Pairwise $\rightarrow$ uniform scoring system along the sequence
- Profiles $\rightarrow$ position specific scoring scheme
- HMMs $\rightarrow$ insertions/deletions modelled separately + variable topology (including simple grammatical structures).

Applications of MCs and HMMs

- bacterial gene finders (MC)
- mRNA splicing (MC)
- trans-membrane helix prediction
- modeling signal peptides
- **modeling families of aligned proteins**
- multiple sequence alignment

Pfam

Pfam is a large collection of multiple sequence alignments and hidden Markov models covering many common protein domains.
Version 5.5, September 2000, 2478 families.
Large coverage of known proteins ($\sim 63\%$).

$\Rightarrow$ www.sanger.ac.uk/Software/Pfam/index.shtml

Web **resources**

HMMER by Sean Eddy,
hmmer.org
SAM from UCSC's Computational Biology Group,
www.cse.ucsc.edu/research/compbio/sam.html

References

Pensez-y!

L'impression de ces notes n'est probablement pas nécessaire!