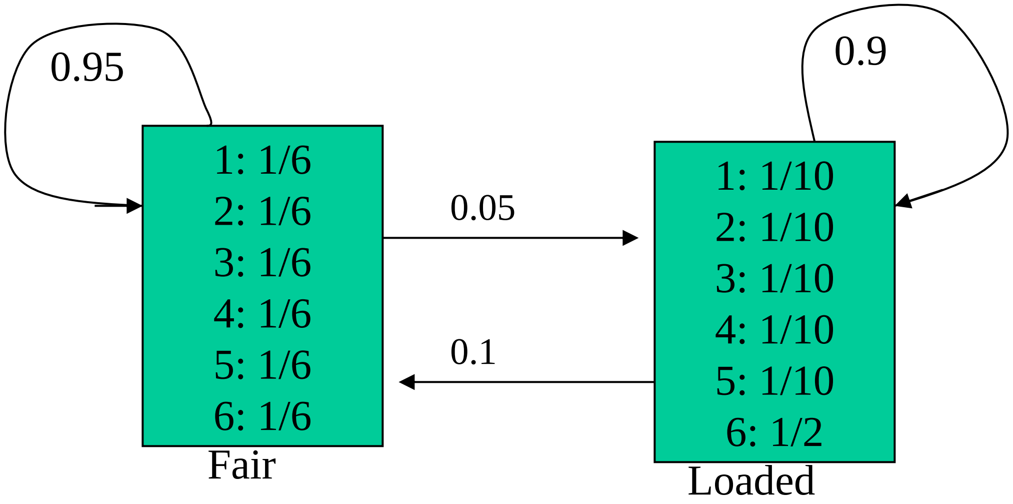
GLOBEX Bioinformatics
(Summer 2015)

Hidden Markov Models (I)

a. The model
b. The decoding: Viterbi algorithm

Hidden Markov models

- A Markov chain of states
- At each state, there are a set of possible observables (symbols), and
- The states are not directly observable, namely, they are hidden.
- E.g., Casino fraud



- Three major problems
  - Most probable state path
  - The likelihood
  - Parameter estimation for HMMs

A biological example: CpG islands

- Higher rate of Methyl-C mutating to T in CpG dinucleotides → generally lower CpG presence in genome, except at some biologically important ranges, e.g., in promoters, -- called CpG islands.
- The conditional probabilities $P_{\pm}(N|N')$are collected from ~ 60,000 bps human genome sequences, + stands for CpG islands and – for non CpG islands.

P_+	A	C	G	T
A	.180	.274	.426	.120
C	.171	.368	**.274**	.188
G	.161	.339	.375	.125
T	.079	.355	.384	.182

P_-	A	C	G	T
A	.300	.205	.285	.210
C	.322	.298	**.078**	.302
G	.248	.246	.298	.208
T	.177	.239	.292	.292

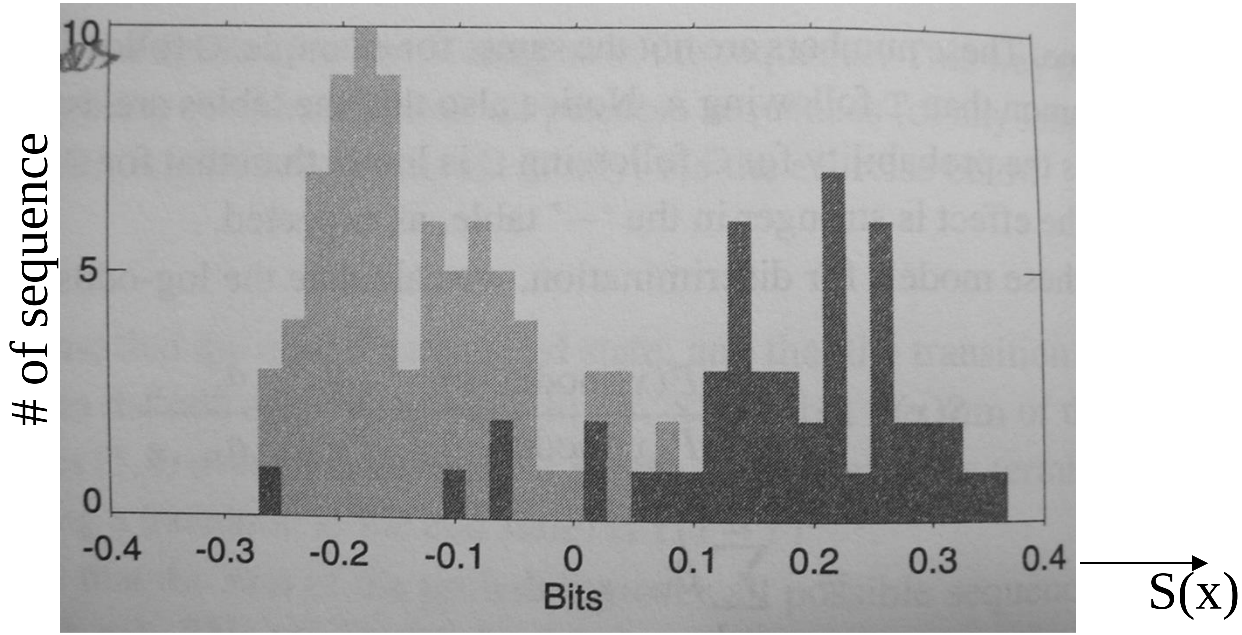
Task 1: given a sequence x, determine if it is a CpG island.

One solution: compute the log-odds ratio scored by the two Markov chains:

$$S(x) = \log [\, P(x \mid \text{model } +) / P(x \mid \text{model } -)]$$

where $P(x \mid \text{model } +) = P_+(x_2|x_1)\, P_+(x_3|x_2) \ldots P_+(x_L|x_{L-1})$ and

$P(x \mid \text{model } -) = P_-(x_2|x_1)\, P_-(x_3|x_2) \ldots P_-(x_L|x_{L-1})$



Histogram of the length-normalized scores
(CpG sequences are shown as dark shaded)

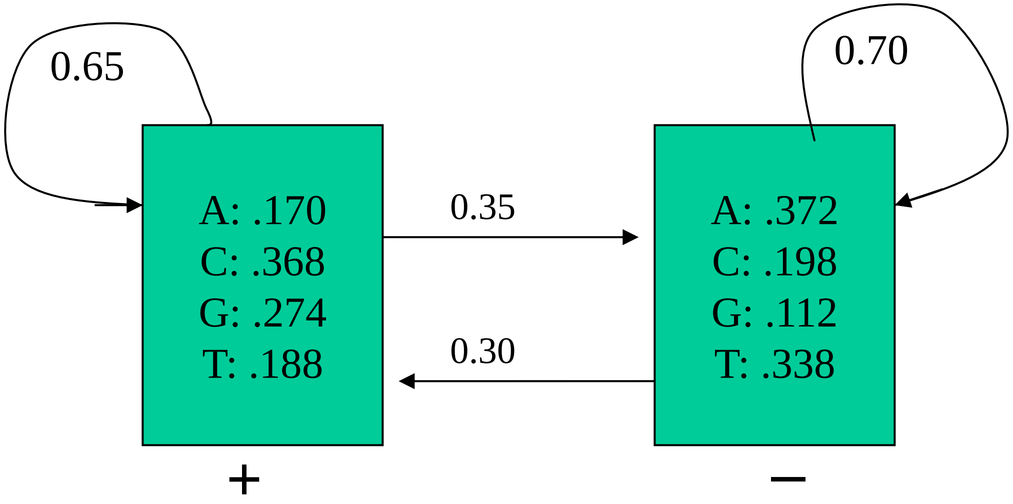
Task 2: For a *long* genomic sequence x, label these CpG islands, if there are any.

Approach 1: Adopt the method for Task 1 by calculating the log-odds score for a window of, say, 100 bps around every nucleotide and plotting it.

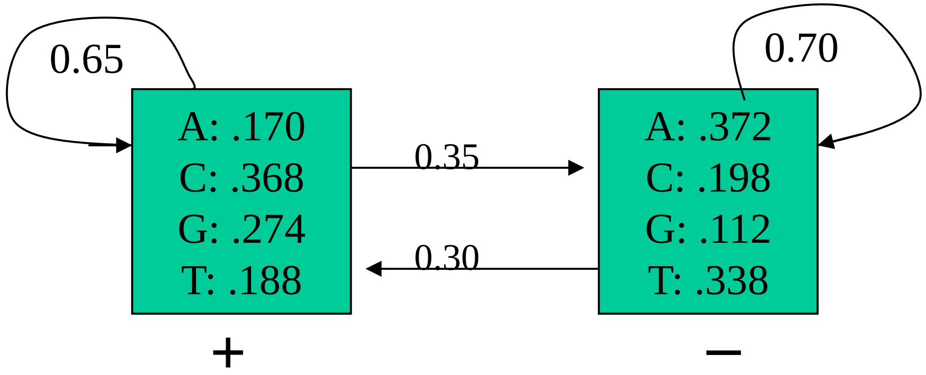
Problems with this approach:

- – Won’t do well if CpG islands have sharp boundary and variable length
- – No effective way to choose a good Window size.

Approach 2: using hidden Markov model



- The model has two states, "+" for CpG island and "-" for non CpG island. Those numbers are made up here, and shall be fixed by learning from training examples.
- The notations: a_{kl} is the transition probability from state k to state l; $e_k(b)$ is the emission frequency – probability that symbol b is seen when in state k.



The probability that sequence x is emitted by a state path π is:

$$P(x, \pi) = \prod_{i=1 \text{ to } L} e_{\pi i}(x_i)\, a_{\pi i\, \pi i+1}$$

```
i:123456789
x:TGCGCGTAC
π:--++++---
```

P(x, π) = 0.338 × 0.70 × 0.112 × 0.30 × 0.368 × 0.65 × 0.274 × 0.65 × 0.368 × 0.65 × 0.274 × 0.35 × 0.338 × 0.70 × 0.372 × 0.70 × 0.198.

Then, the probability to observe sequence x in the model is

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

which is also called the likelihood of the model.

Decoding: Given an observed sequence x, what is the most probable state path, i.e.,

$$\pi^* = \text{argmax}_{\pi}\ P(x, \pi)$$

Q: Given a sequence x of length L, how many state paths do we have?
A: N^L, where N stands for the number of states in the model.
As an exponential function of the input size, it precludes enumerating all possible state paths for computing P(x).

Let $v_k(i)$ be the probability for the most probable path ending at position i with a state k.

Viterbi Algorithm

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

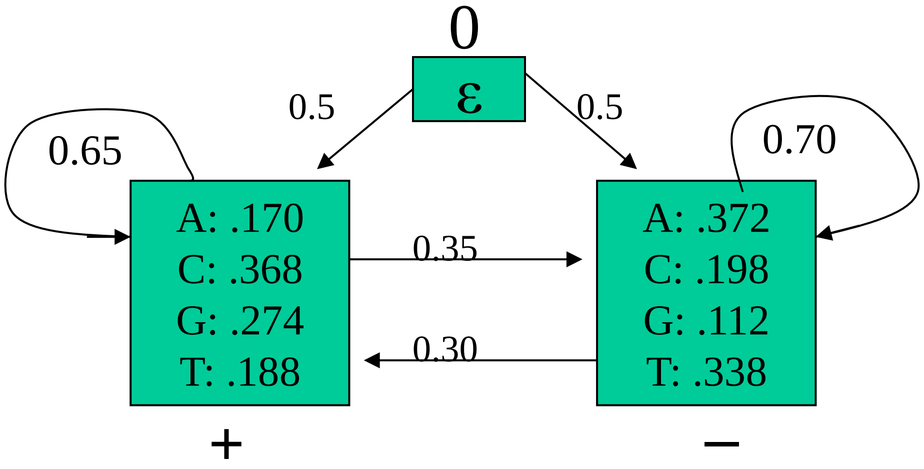
Recursion: $v_k(i) = e_k(x_i) \max_j (v_j(i-1) a_{jk})$;

$ptr_i(k) = \operatorname{argmax}_j (v_j(i-1) a_{jk})$;

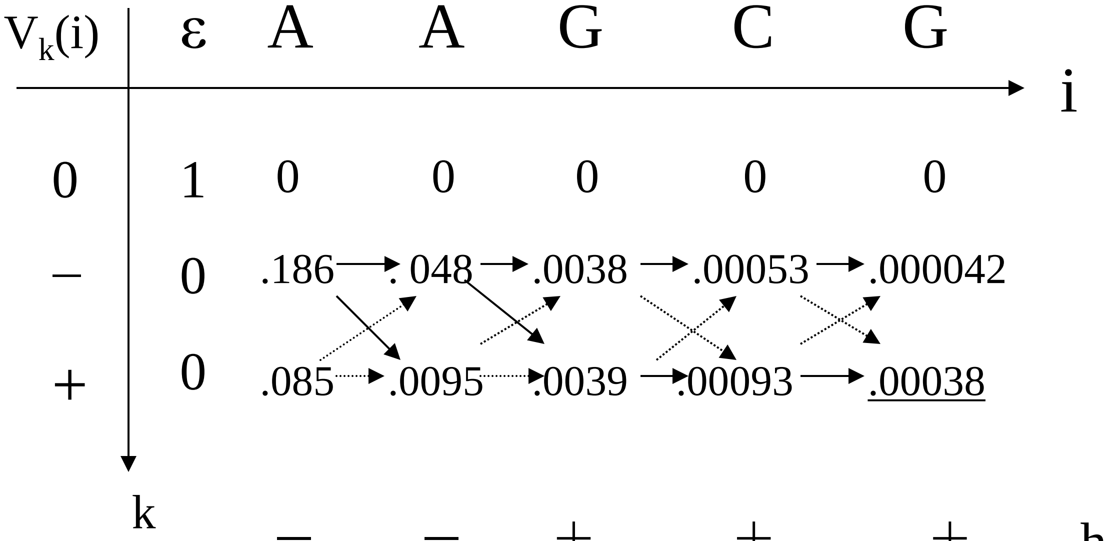
Termination: $P(x, \pi^*) = \max_k (v_k(L) a_{k0})$;

$\pi^*_L = \operatorname{argmax}_j (v_j(L) a_{j0})$;

Traceback: $\pi^*_{i-1} = ptr_i (\pi^*_i)$.



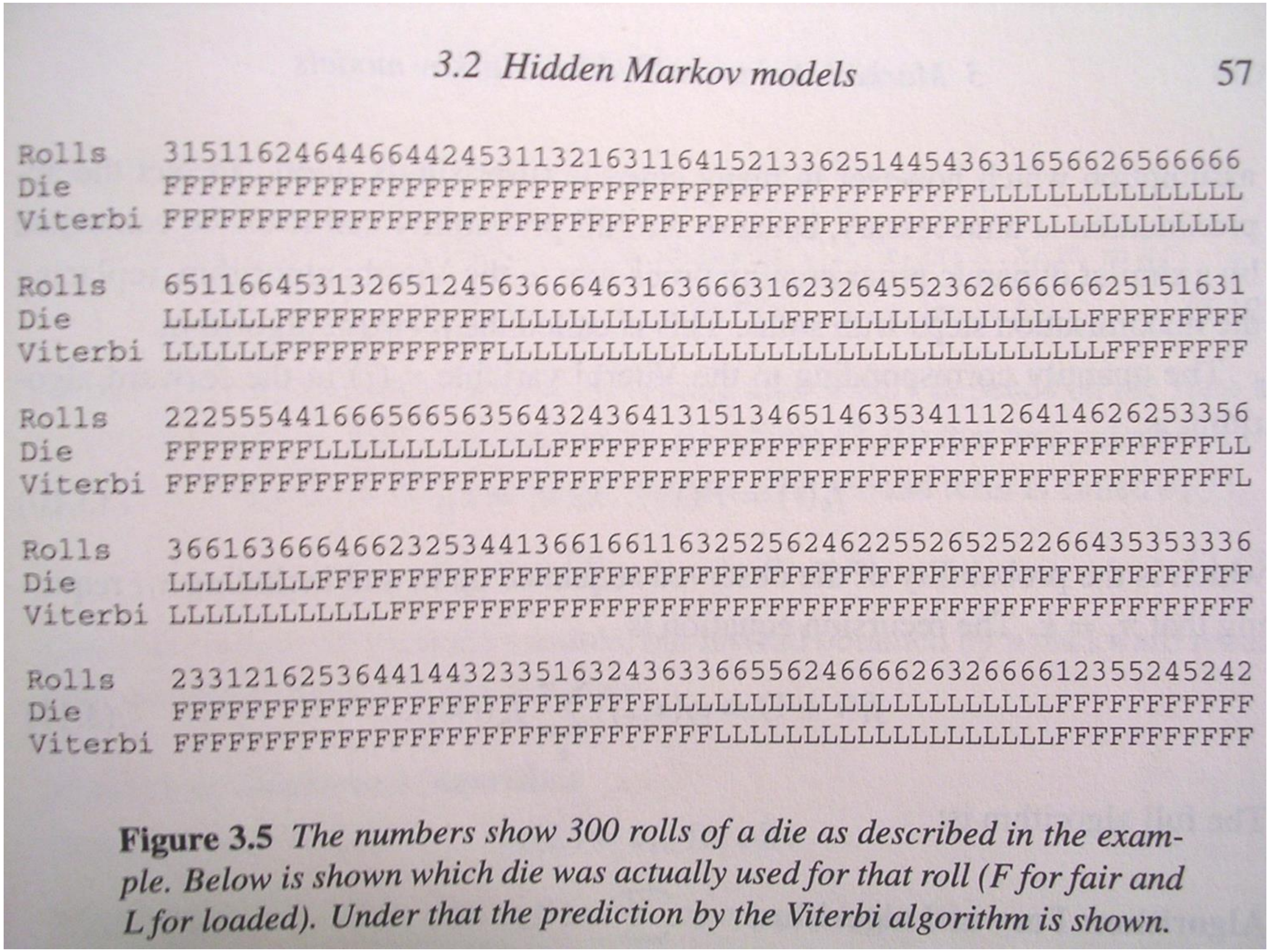
$v_k(i) = e_k(x_i) \max_j (v_j(i-1)\, a_{jk});$



$V_k(i)$	ε	A	A	G	C	G
0	1	0	0	0	0	0
–	0	.186	. 048	.0038	.00053	.000042
+	0	.085	.0095	.0039	.00093	.00038

	–	–	+	+	+	hidden state path

Casino Fraud: investigation results by Viterbi decoding



```
Rolls   315116246446644245311321631164152133625144543631656626566666
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLL

Rolls   651166453132651245636664631636663162326455236266666625151631
Die     LLLLLLFFFFFFFFFFFFLLLLLLLLLLLLLLLLFFFLLLLLLLLLLLLLLFFFFFFFFF
Viterbi LLLLLLFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLFFFFFFFF

Rolls   222555441666566563564324364131513465146353411126414626253356
Die     FFFFFFFFLLLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL

Rolls   366163666466232534413661661163252562462255265252266435353336
Die     LLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Viterbi LLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Rolls   233121625364414432335163243633665562466662632666612355245242
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFF
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFF
```

Figure 3.5 *The numbers show 300 rolls of a die as described in the example. Below is shown which die was actually used for that roll (F for fair and L for loaded). Under that the prediction by the Viterbi algorithm is shown.*

- The log transformation for Viterbi algorithm

$$v_k(i) = e_k(x_i) \max_j (v_j(i-1)\, a_{jk});$$

$$\underline{\boldsymbol{a}}_{jk} = \log a_{jk};$$

$$\underline{\boldsymbol{e}}_k(x_i) = \log e_k(x_i);$$

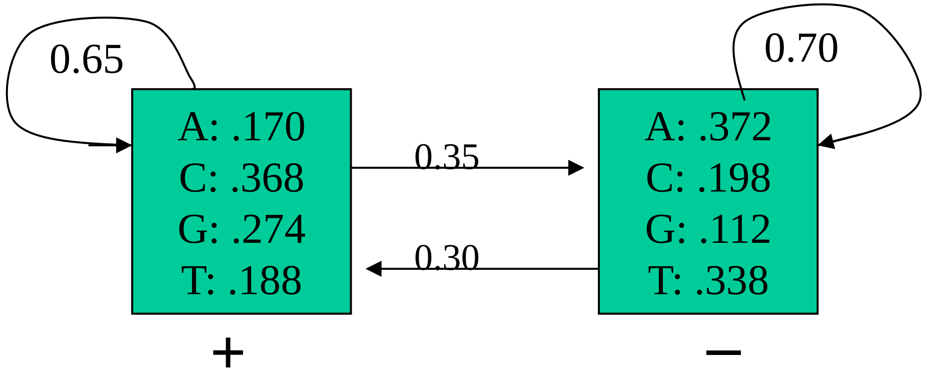
$$\underline{\boldsymbol{v}}_k(i) = \log v_k(i);$$

$$\underline{\boldsymbol{v}}_k(i) = \underline{\boldsymbol{e}}_k(x_i) + \max_j (\underline{\boldsymbol{v}}_j(i-1) + \underline{\boldsymbol{a}}_{jk});$$

GLOBEX Bioinformatics (Summer 2015)

Hidden Markov Models (II)

- The model likelihood: Forward algorithm, backward algorithm
- Posterior decoding



The probability that sequence x is emitted by a state path π is:

$$P(x, \pi) = \prod_{i=1 \text{ to } L} e_{\pi i}(x_i)\, a_{\pi i\, \pi i+1}$$

```
i:123456789
x:TGCGCGTAC
π:--++++---
```

$P(x, \pi) = 0.338 \times 0.70 \times 0.112 \times 0.30 \times 0.368 \times 0.65 \times 0.274 \times 0.65 \times 0.368 \times 0.65 \times 0.274 \times 0.35 \times 0.338 \times 0.70 \times 0.372 \times 0.70 \times 0.198.$

Then, the probability to observe sequence x in the model is

$$P(x) = \Sigma_\pi P(x, \pi),$$

which is also called the likelihood of the model.

How to calculate the probability to observe sequence x in the model?

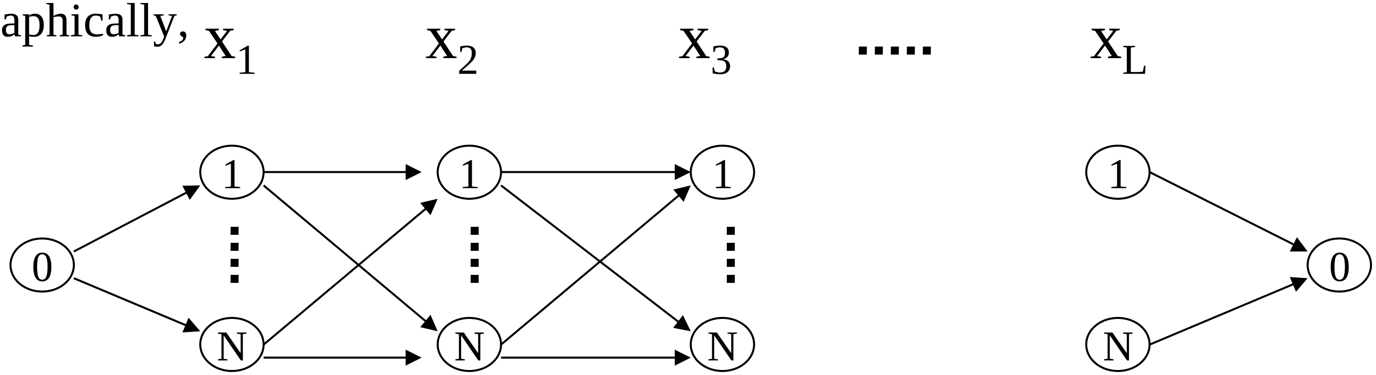
$$P(x) = \Sigma_\pi P(x, \pi)$$

Let $f_k(i)$ be the probability contributed by all paths from the beginning up to (and include) position i with the state at position i being k.

The the following recurrence is true:

$$\mathbf{f_{k}(i) = [\Sigma_{j} f_{j}(i\text{-}1)\, a_{jk}]\, e_k(x_i)}$$

Graphically,



Again, a silent state 0 is introduced for better presentation

Forward algorithm

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

Recursion: $f_k(i) = e_k(x_i) \Sigma_j f_j(i-1) a_{jk}$.

Termination: $P(x) = \Sigma_k f_k(L) a_{k0}$.

Time complexity: $O(N^2L)$, where N is the number of states and L is the sequence length.

Let $b_k(i)$ be the probability contributed by all paths that pass state k at position i.

$b_k(i) = P(x_{i+1}, \ldots, x_L \mid \pi(i) = k)$

Backward algorithm

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

Recursion ($i = L-1, \ldots, 1$): $b_k(i) = \Sigma_j \, a_{kj} \, e_j(x_{i+1}) \, b_j(i+1)$.

Termination: $P(x) = \Sigma_k \, a_{0k} \, e_k(x_1) b_k(1)$.

Time complexity: $O(N^2L)$, where N is the number of states and L is the sequence length.

Posterior decoding

$$P(\pi_i = k \mid x) = P(x, \pi_i = k) / P(x) = f_k(i) b_k(i) / P(x)$$

Algorithm:

```
for i = 1 to L
        do argmax_k $P(\pi_i = k \mid x)$
```

Notes: 1. Posterior decoding may be useful when there are multiple almost most probable paths, or when a function is defined on the states.

2. The state path identified by posterior decoding may not be most probable overall, or may not even be a viable path.

GLOBEX Bioinformatics
(Summer 2015)

Hidden Markov Models (III)

- Viterbi training
- Baum-Welch algorithm
- Maximum Likelihood
- Expectation Maximization

Model building

- Topology
  - Requires domain knowledge
- Parameters
  - When states are labeled for sequences of observables
    - Simple counting:

$$a_{kl} = A_{kl} / \Sigma_{l'} A_{kl'} \text{ and } e_k(b) = E_k(b) / \Sigma_{b'} E_k(b')$$

  - When states are not labeled

Method 1 (Viterbi training)

1. Assign random parameters
2. Use Viterbi algorithm for labeling/decoding
2. Do counting to collect new a_{kl} and $e_k(b)$;
3. Repeat steps 2 and 3 until stopping criterion is met.

Method 2 (Baum-Welch algorithm)

Baum-Welch algorithm (Expectation-Maximization)

- An iterative procedure similar to Viterbi training
- Probability that a_{kl} is used at position i in sequence j.

  $P(\pi_i = k, \pi_{i+1} = l \mid x, \theta) = f_k(i)\, a_{kl}\, e_l(x_{i+1})\, b_l(i+1) / P(x^j)$

Calculate the expected number of times that is used by summing over all position and over all training sequences.

$$A_{kl} = \Sigma_j \{(1/P(x^j)\ [\Sigma_i\, f_k^j(i)\, a_{kl}\, e_l(x^j_{i+1})\, b_l^j(i+1)]\ \}$$

Similarly, calculate the expected number of times that symbol b is emitted in state k.

$$E_k(b) = \Sigma_j \{(1/P(x^j)\ [\Sigma_{\{i|x\_i\wedge j = b\}}\, f_k^j(i)\, b_k^j(i)]\ \}$$

Maximum Likelihood

Define $L(\theta) = P(x|\ \theta)$

Estimate θ such that the distribution with the estimated θ best agrees with or support the data observed so far.

$$\theta^{ML} = \underset{\theta}{\text{argmax}}\ L(\theta)$$

E.g. There are red and black balls in a box. What is the probability P of picking up a black ball?

Do sampling (with replacement).

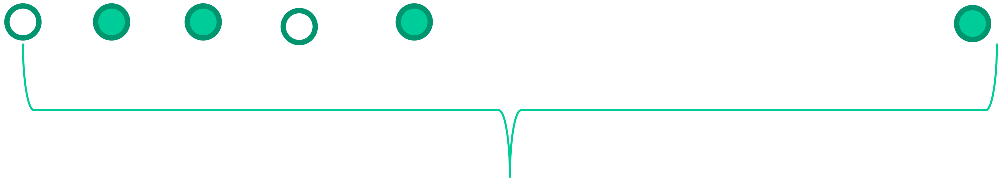
Maximum Likelihood

Define L(θ) = P(x| θ)
Estimate such that the distriibution with the estimated best agrees with or supports the data observed so far.

$\theta^{ML} = \text{argmax}\ \theta\ L(\theta)$
When L(θ) is differentiable,

$$\frac{\partial L(\theta)}{\partial \theta}\Big|_{\theta^{ML}} = 0$$

For example, want to know the ratio: # of blackball/# of whiteball, in other words, the probability P of picking up a black ball. Sampling (with replacement):



100 times

Counts: whiteball 91, blackball 9

Prob (iid) = $p^9 (1-p)^{91}$
Likelihood $L(p) = p^9(1-p)^{91}$.

$$\frac{\partial L(p)}{\partial P} = 9p^8(1-p)^{91} - 91p^9(1-p)^{90} = 0$$

=> $P^{ML} = 9/100 = 9\%$.   The ML estimate of P is just the frequency.

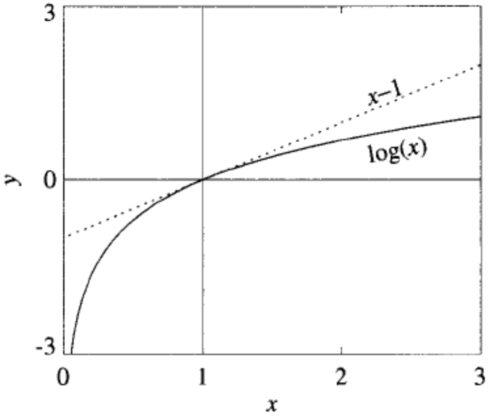
A proof that the observed frequency -> ML estimate of probabilities for polynomial distribution

Let Counts n_i for outcome i

The observed frequencies $\theta_i = n_i / N$, where $N = \sum_i n_i$

If $\theta_i^{ML} = n_i / N$, then $P(n|\theta^{ML}) > p(n|\theta)$ for any $\theta \neq \theta^{ML}$

Proof:



$$\log \frac{P(n|\theta^{ML})}{P(n|\theta)} = \log \frac{\prod_i (\theta_i^{ML})^{n_i}}{\prod_i (\theta_i)^{n_i}} = \log \prod_i \left(\frac{\theta_i^{ML}}{\theta_i}\right)^{n_i}$$

$$= \sum_i n_i \log\left(\frac{\theta_i^{ML}}{\theta_i}\right) = \mathrm{N} \sum_i \frac{n_i}{N} \log\left(\frac{\theta_i^{ML}}{\theta_i}\right) = \sum_i \theta_i^{ML} \log\left(\frac{\theta_i^{ML}}{\theta_i}\right)$$

$$= \mathrm{H}(\theta^{ML} \,||\, \theta) \geq 0$$

Maximum Likelihood: pros and cons

- Consistent, i.e., in the limit of a large amount of data, ML estimate converges to the true parameters by which the data are created.
- Simple
- Poor estimate when data are insufficient.

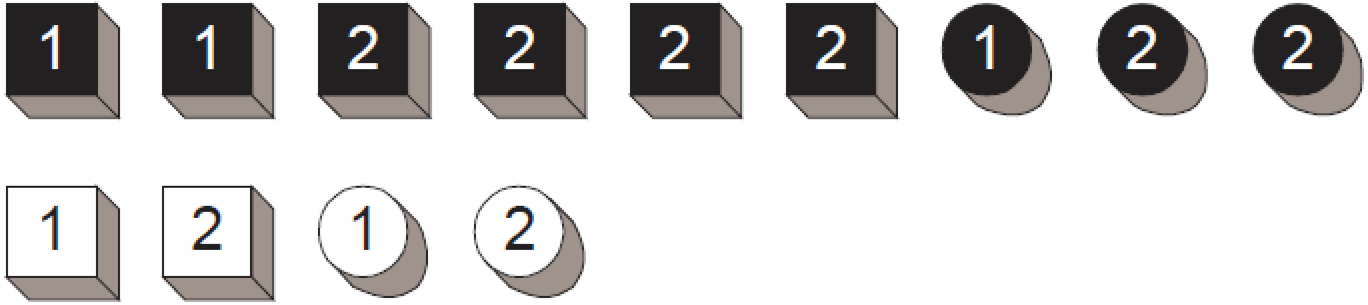
  e.g., if you roll a die for less than 6 times, the ML estimate for some numbers would be zero.

Pseudo counts:

$$\theta_i = \frac{n_i + \alpha_i}{N + A},$$

where A = $\sum_i \alpha_i$

Conditional Probability and Join Probability



P(one) = 5/13

P(square) = 8/13

P(one, square) = 3/13

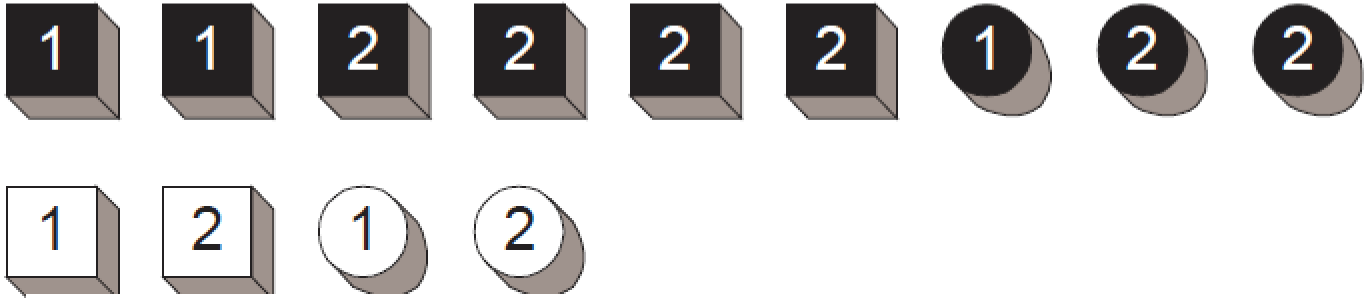
P(one | square) = 3/8 = P(one, square) / P(square)

In general, P(D,M) = P(D|M)P(M) = P(M|D)P(D)

=> **Baye's Rule:** $P(\mathrm{M} \mid \mathrm{D}) = \dfrac{P(D \mid M)P(M)}{P(D)}$

$$\begin{aligned} P(\text{One}|\text{Black}) &= \frac{P(\text{Black}|\text{One})P(\text{One})}{P(\text{Black}|\text{One})P(\text{One}) + P(\text{Black}|\text{Two})P(\text{Two})} \\ &= \frac{(\frac{3}{5})(\frac{5}{13})}{(\frac{3}{5})(\frac{5}{13}) + (\frac{6}{8})(\frac{8}{13})} = \frac{1}{3}, \end{aligned}$$

Conditional Probability and Conditional Independence



$$
\begin{aligned}
P(\text{One}) &= \frac{5}{13} \\
P(\text{One}|\text{Square}) &= \frac{3}{8}
\end{aligned}
$$

$$
\begin{aligned}
P(\text{One}|\text{Black}) &= \frac{3}{9} = \frac{1}{3} \\
P(\text{One}|\text{Square} \cap \text{Black}) &= \frac{2}{6} = \frac{1}{3}
\end{aligned}
$$

$$
\begin{aligned}
P(\text{One}|\text{White}) &= \frac{2}{4} = \frac{1}{2} \\
P(\text{One}|\text{Square} \cap \text{White}) &= \frac{1}{2}.
\end{aligned}
$$

So One *and* Square *are not independent, but they are conditionally independent given* Black *and given* White.

Baye’s Rule:

$$P(\mathrm{M} \mid \mathrm{D}) = \frac{P(D \mid M)P(M)}{P(D)}$$

Example: disease diagnosis/inference

P(Leukemia | Fever) = ?

P(Fever | Leukemia) = 0.85

P(Fever) = 0.9

P(Leukemia) = 0.005

P(Leukemia | Fever) = P(F|L)P(L)/P(F) = 0.85*0.01/0.9 = 0.0047

Bayesian Inference

Maximum a posterior estimate

$$\theta^{MAP} = \underset{\theta}{\arg\max}\, P(\theta \mid \mathrm{x})$$

Expectation Maximization

$$\mathrm{P}(x, y \mid \theta) = P(y \mid x, \theta) P(\mathrm{x} \mid \theta)$$

$$P(x \mid \theta) = P(x, y \mid \theta) \,/\, P(y \mid x, \theta)$$

$$\log P(x \mid \theta) = \log P(x, y \mid \theta) - \log P(\mathrm{y} \mid \mathrm{x}, \theta)$$

$$\sum_y P(y \mid x, \theta^t) \quad (\qquad\qquad\qquad\qquad\qquad\qquad)$$

Expectation

$$\log P(x \mid \theta) = \sum_y P(y \mid x, \theta^t) \log P(x, y \mid \theta) - \sum_y P(y \mid x, \theta^t) \log P(y \mid \mathrm{x}, \theta)$$

$$Q(\theta \mid \theta^t) = \sum_y P(y \mid x, \theta^t) \,\mathrm{logP}(\mathrm{x}, \mathrm{y} \mid \theta)$$

$$\log P(x \mid \theta) - \log P(x \mid \theta^t)$$

$$= Q(\theta \mid \theta^t) - Q(\theta^t \mid \theta^t) + \sum_y P(y \mid x, \theta^t) \log \frac{P(y \mid x, \theta^t)}{P(y \mid x, \theta)}$$

$$\geq Q(\theta \mid \theta^t) - Q(\theta^t \mid \theta^t)$$

$$\theta^{t+1} = \arg\max_{\theta} Q(\theta \mid \theta^t)$$

Maximization

EM explanation of the Baum-Welch algorithm

We like to maximize by choosing θ

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

But state path π is hidden variable. Thus, EM.

$$Q(\theta \mid \theta^{t}) = \sum_{\pi} P(\pi \mid x, \theta^{t}) \log P(x, \pi \mid \theta)$$

$$P(x,\pi \mid \theta) = \prod_{k=1}^{M} \prod_{b} [e_k(b)]^{E_k(b,\pi)} \prod_{k=0}^{M} \prod_{l=1}^{M} a_{kl}^{A_{kl}(\pi)},$$

$$Q(\theta \mid \theta^{t}) = \sum_{\pi} P(\pi \mid x, \theta^{t}) \times$$

$$\left[\sum_{k=1}^{M} \sum_{b} E_k(b,\pi) \log e_k(b) + \sum_{k=0}^{M} \sum_{l=1}^{M} A_{kl}(\pi) \log a_{kl} \right].$$

EM Explanation of the Baum-Welch algorithm

$$E_k(b) = \sum_{\pi} P(\pi | x, \theta^t) E_k(b, \pi) \quad \text{and} \quad A_{kl} = \sum_{\pi} P(\pi | x, \theta^t) A_{kl}(\pi).$$

$$Q(\theta | \theta^t) = \sum_{k=1}^{M} \sum_{b} E_k(b) \log e_k(b) + \sum_{k=0}^{M} \sum_{l=1}^{M} A_{kl} \log a_{kl}.$$

E-term A-term

A-term is maximized if $a_{kl}^{EM} = \dfrac{A_{kl}}{\sum_{l'} A_{kl'}}$

E-term is maximized if $e_k^{EM}(b) = \dfrac{E_k(b)}{\sum_{b'} E_k(b')}$

GLOBEX Bioinformatics (Summer 2015)

Hidden Markov Models (IV)

a. Profile HMMs

b. ipHMMs

c. GeneScan

d. TMMOD

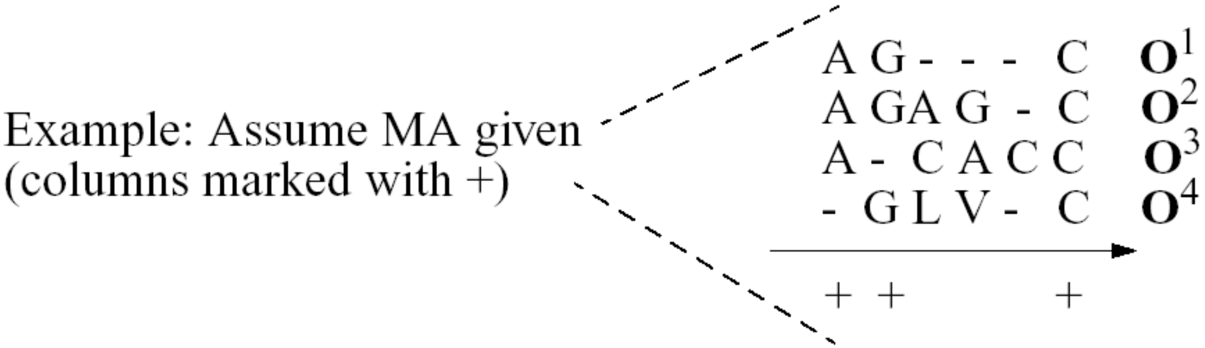
Profile HMM for a family of sequences

Applications of HMM’s

- Given a family of sequences, $\mathbf{O}^l = O^l_1 \ldots O^l_{K^l}$, build a hidden Markov model that best fits to this family-->Problem 3
    - Correct multiple alignment is given--> Problem 3, path known
        - MA built from structural information
        - MA obtained from other sequence based alignment procedures
    - Alignment is not assumed--> Problem 3, path not known (B-W)
- Use the obtained model to:
    - Score potential matches of new sequences-->Problem 1
    - Align new sequences--> Problem 2

Profile HMM: Correct alignment assumed

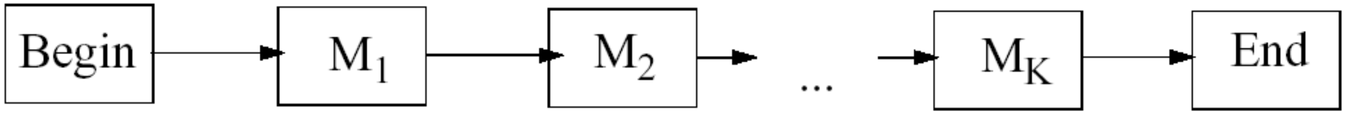
HMM construction



Example: Assume MA given
(columns marked with +)

A	G	-	-	-	C	$\mathbf{O}^1$
A	G	A	G	-	C	$\mathbf{O}^2$
A	-	C	A	C	C	$\mathbf{O}^3$
-	G	L	V	-	C	$\mathbf{O}^4$
+	+				+	

- Segments of family where an alignment exists are produced by MATCH STATES



- Generation probabilities are position dependent!
- In previous example, K=3

Profile HMM: Correct alignment assumed

- Handling insertions: Portion of the sequences that are not aligned ---> Add INSERT STATES

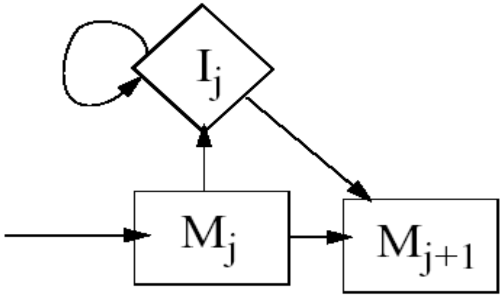
Example: Assume MA given (columns marked with +)

```
A G - - -   C    O^1
A G A G -   C    O^2
A - C A C   C    O^3
- G L V -   C    O^4

+ +         +
```

- To cope with all possibilities for insertions, an insert state should be added after each match state

State I_k inserts sequence just after match state M_k (i.e., aligned column k)



$\mathbf{O}^1$--> $M_1M_2M_3$

$\mathbf{O}^2$--> $M_1M_2I_2I_2M_3$

$\mathbf{O}^3$--> M_1? State M_2 is skipped

Profile HMM: Correct alignment assumed

- Handling deletions: Portion of the sequences that “skips” the alignment---> Add SILENT (DELETE) STATES

Example: Assume MA given (columns marked with +)

A G - - - C $\mathbf{O}^1$

A G A G - C $\mathbf{O}^2$

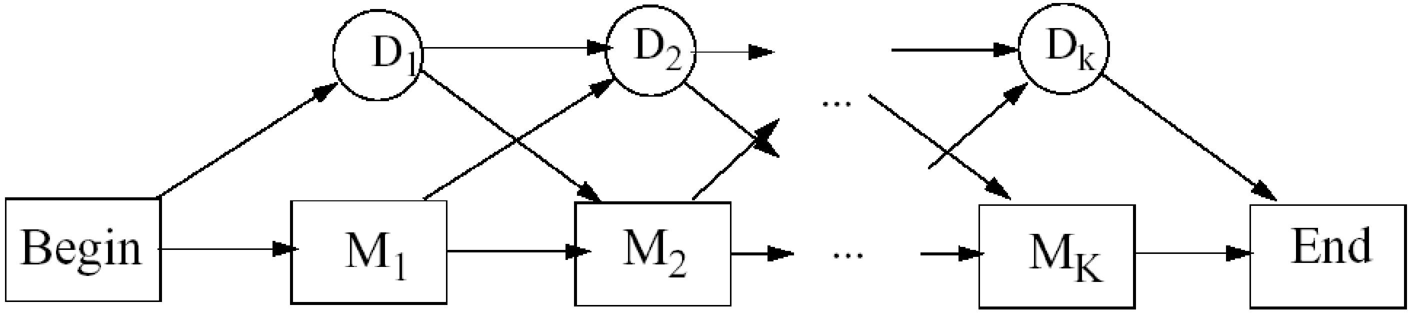
A $\boxed{-}$ C A C C $\mathbf{O}^3$

$\boxed{-}$ G L V - C $\mathbf{O}^4$

+ \+ \+

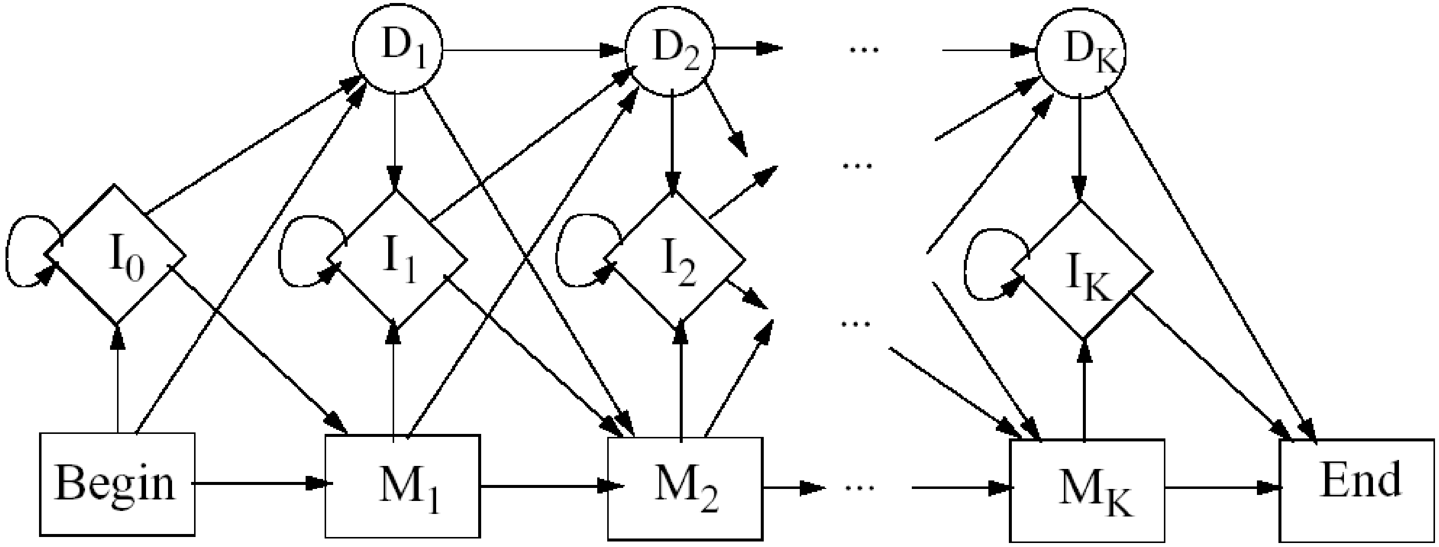
- To cope with all possibilities for deletions
  - Connect all possible match states (big complexity)
  - Add silent states (less complexity, but loss of generality)-->NO EMISSION

State D_k skips match state M_k (i.e., aligned column k)



Profile HMM: Correct alignment assumed

Resulting HMM (Profile HMM)



- Notice we have added transitions between insert and delete states

Example: Assume MA given (columns marked with +)

A G - - - C $\mathbf{O}^1$ $M_1M_2M_3$
A G A G - C $\mathbf{O}^2$ $M_1M_2I_2I_2M_3$
A - C A C C $\mathbf{O}^3$ $M_1D_2I_2I_2I_2M_3$
- G L V - C $\mathbf{O}^4$ $D_1M_2I_2I_2M_3$

→

+ \+ \+

Profile HMM: Correct alignment assumed

Key idea of profile HMM

- Transition and emission probabilities capture specific information about each position in the multiple alignment of the whole family
- Profile HMM=Statistical model representing the family

Questions

- How do we build the profile HMM that best fits to a given family? -->Problem 3 (simplified)
- How do we detect potential membership in this family (for new sequences)? --> Problem 1
- How do we align a new sequence? --> Problem 2

Parameterization of profile HMM's: Correct alignment assumed

Profile HMM parametrization (simplified Problem 3)

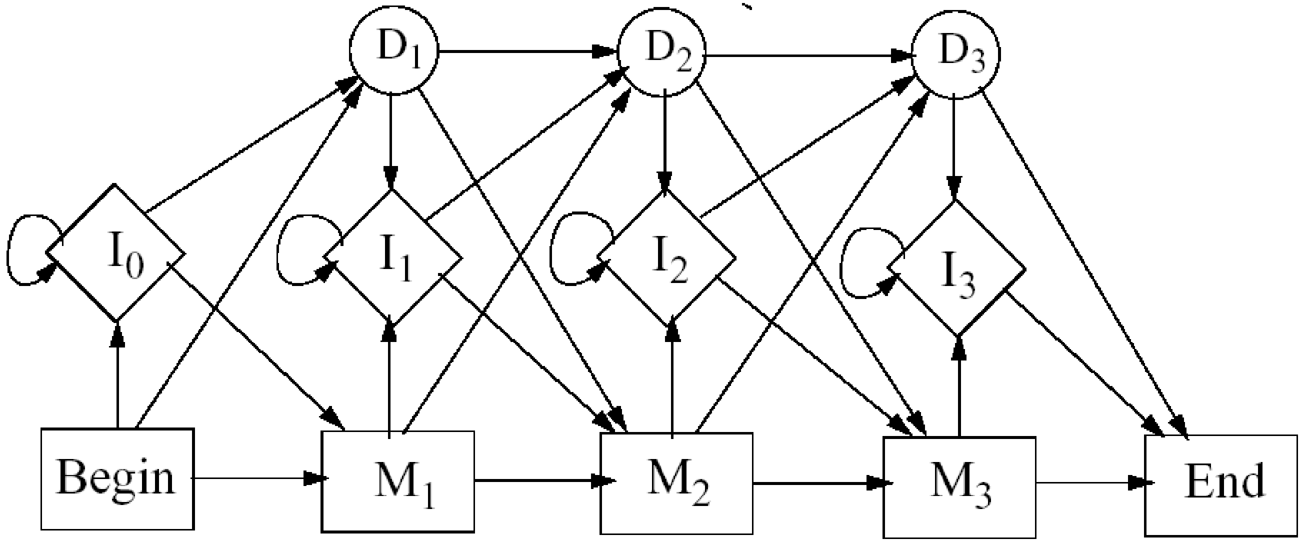
- **Model length**
  - Length (and structure) completely defined when we decide which MA columns should be assigned to match states
    - Manual construction
    - Heuristic construction: e.g., column aligned if proportion of gaps is less than a threshold
    - More sophisticated methods
- **Parameter estimation**
  - Alignment is given-->Path through model is given for any sequence
  - Apply solution to Problem 3 when path is given (just count events)

Parameterization of profile HMM's: Correct alignment assumed

Previous example

MA given
(columns marked with +)

A G - - - C	$\mathbf{O}^1$	$M_1M_2M_3$
A G A G - C	$\mathbf{O}^2$	$M_1M_2I_2I_2M_3$
A - C A C C	$\mathbf{O}^3$	$M_1D_2I_2I_2I_2M_3$
- G L V - C	$\mathbf{O}^4$	$D_1M_2I_2I_2M_3$
+ + +		



Parameterization of profile HMM's: Correct alignment assumed

Emission probabilities: Estimate from number of emissions

$N(A|M_1)=3$ $N(\text{other}|M_1)=0$

$N(A|M_2)=3$ $N(\text{other}|M_2)=0$

$N(C|M_3)=4$ $N(\text{other}|M_3)=0$

I_0, I_1, I_3 are not used

$N(A|I_2)=2$ $N(C|I_2)=2$ $N(G|I_2)=1$

$N(L|I_2)=1$ $N(V|I_2)=1$ $N(\text{other}|I_2)=0$

Transition probabilities: Estimate from number of transitions

$N(M_1|B)=3$ $N(D_1|B)=1$

$N(M_2|M_1)=3$ $N(D_2|M_1)=1$

$N(M_3|M_2)=1$ $N(I_2|M_2)=2$

$N(E|M_3)=3$

$N(I_2|D_2)=1$

$N(I_2|I_2)=4$ $N(M_3|I_2)=3$

- **If number of sequences is not high enough, estimation should be modified**

Membership in a profile HMM

Detection of potential membership, for a new sequence, in family defined by a profile HMM (Problem 1)

- Apply forward equation
- Since P(**O**|M) is length dependent, usually scoring function is modified

$$\text{Scoring}=\log\frac{P(\mathbf{O}|M)}{P(\mathbf{O}|S)}$$

S is called "standard model": Model to use if sequences were independently distributed

- Other statistical approaches can also be used to improve the scoring system

Multiple alignment using profile HMM's

No alignment is assumed

- From an initially unaligned family of sequences, jointly perform:
  - Profile HMM estimation
  - Alignment estimation

1. Initialization

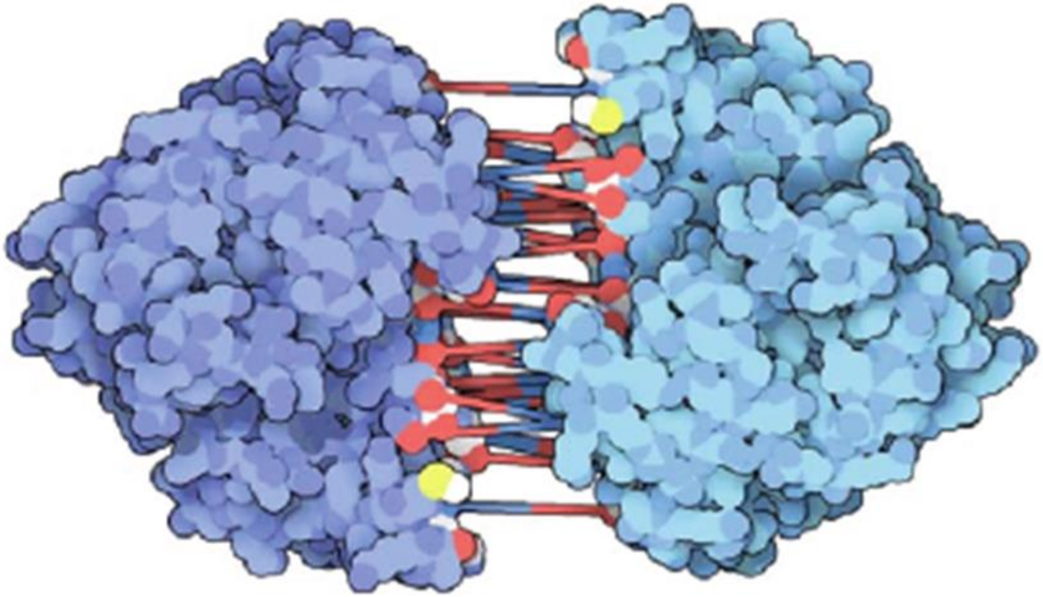
- Choose length of profile HMM and initialize parameters

2. Training

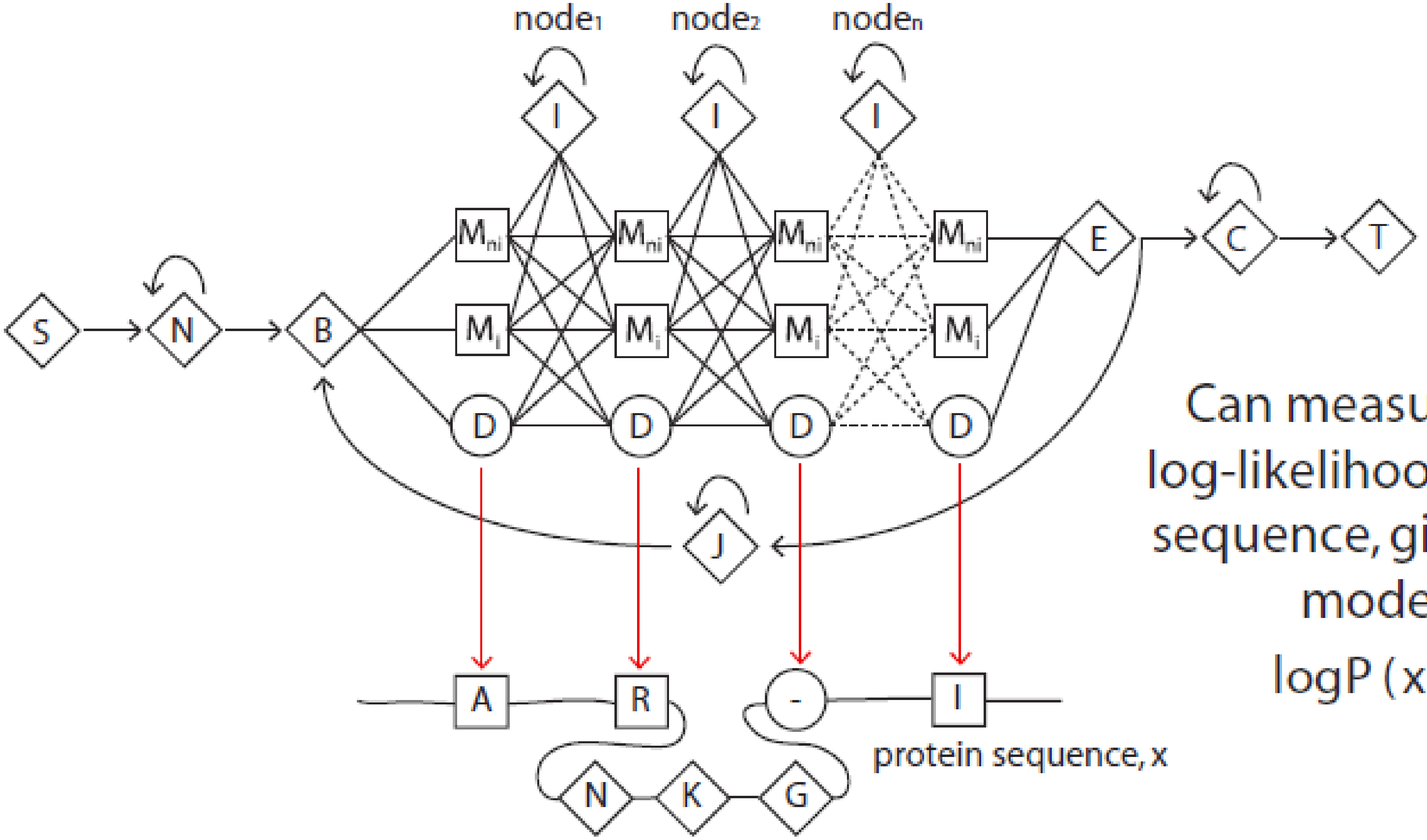
- Estimate parameters of the profile HMM
- Path not known (no alignment)--> Problem 3 (Baum-Welch)

3. Alignment

- Align all sequences using Viterbi algorithm (Problem 2)



Interaction profile HMM (ipHMM)



Can measure the log-likelihood of the sequence, given the model:

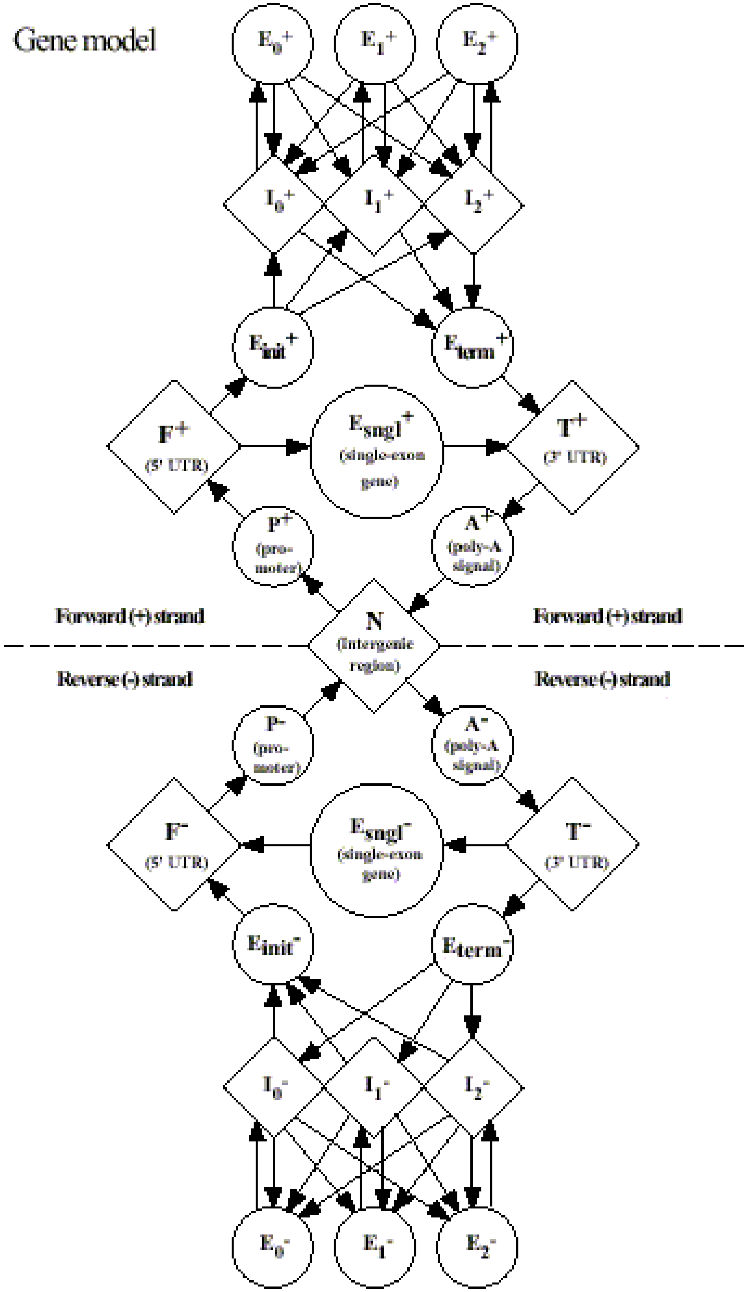
$\log P(x|\theta)$

Friedrich et al, *Bioinformatics* 2006

GENSCAN (generalized HMMs)

- Chris Burge, PhD Thesis ’97, Stanford
- http://genes.mit.edu/GENSCAN.html
- Four components
  - A vector π of initial probabilities
  - A matrix T of state transition probabilities
  - A set of length distribution f
  - A set of sequence generating models P
- Generalized HMMs:
  - at each state, emission is not symbols (or residues), rather, it is a fragment of sequence.
  - Modified viterbi algorithm

Gene model


E0+
E1+
E2+
I0+
I1+
I2+
Einit+
Eterm+
F+
(5' UTR)
Esngl+
(single-exon
gene)
T+
(3' UTR)
P+
(pro-
moter)
A+
(poly-A
signal)
Forward (+) strand
N
(Intergenic
region)
Forward (+) strand
Reverse (-) strand
Reverse (-) strand
P-
(pro-
moter)
A-
(poly-A
signal)
F-
(5' UTR)
Esngl-
(single-exon
gene)
T-
(3' UTR)
Einit-
Eterm-
I0-
I1-
I2-
E0-
E1-
E2-

- Initial state probabilities
  - As frequency for each functional unit to occur in actual genomic data. E.g., as ~ 80% portion are non-coding intergenic regions, the initial probability for state N is 0.80
- Transition probabilities
- State length distributions

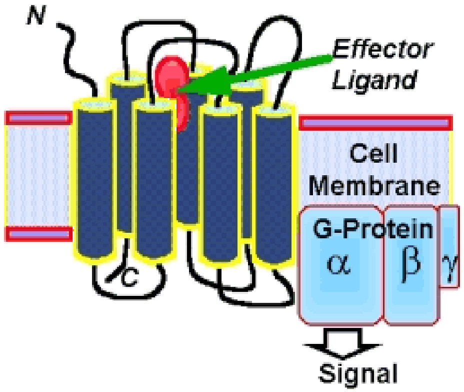
- Training data
  - 2.5 Mb human genomic sequences
  - 380 genes, 142 single-exon genes, 1492 exons and 1254 introns
  - 1619 cDNAs

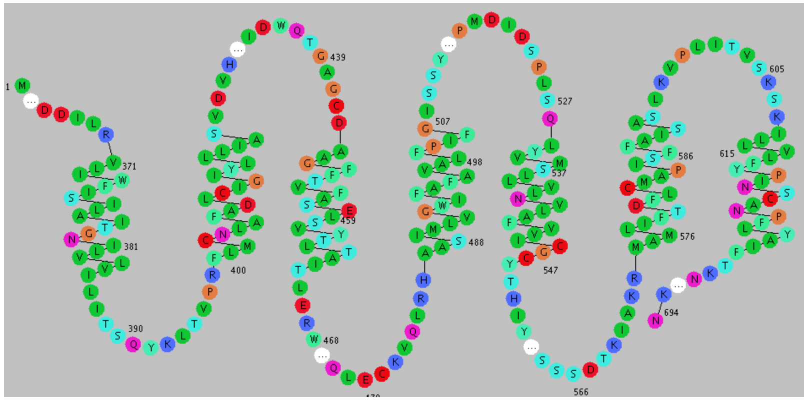
Open areas for research

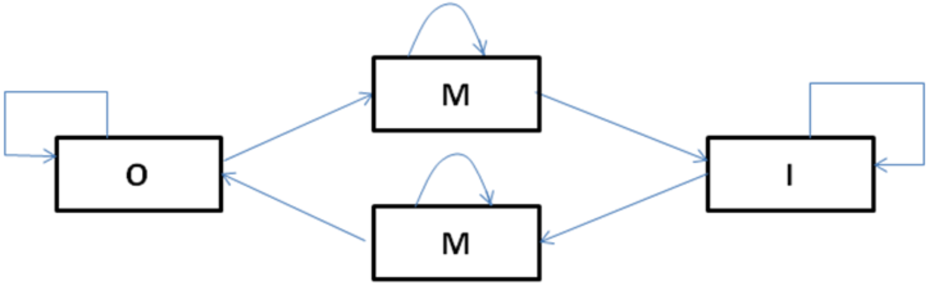
- Model building
  - Integration of domain knowledge, such as structural information, into profile HMMs
  - Meta learning?
- Biological mechanism

  DNA replication
- Hybrid models
  - Generalized HMM
  - …

TMMOD: An improved hidden Markov model for predicting transmembrane topology

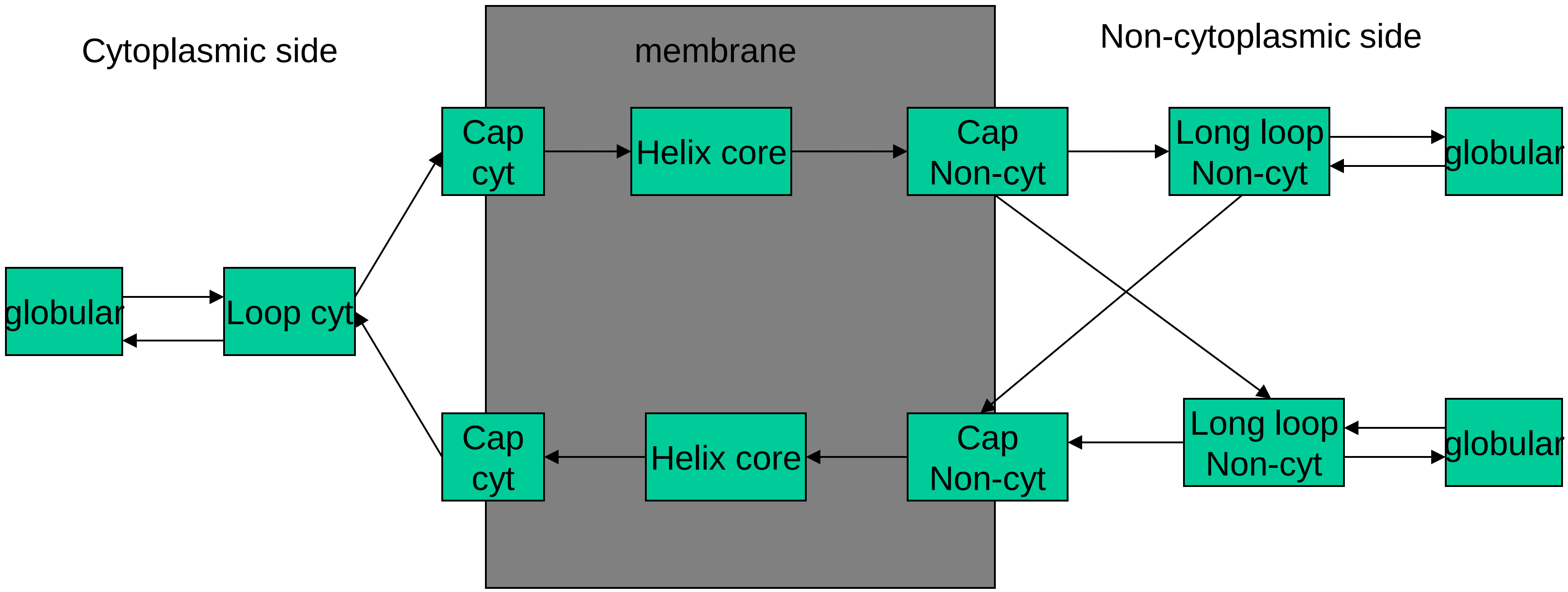




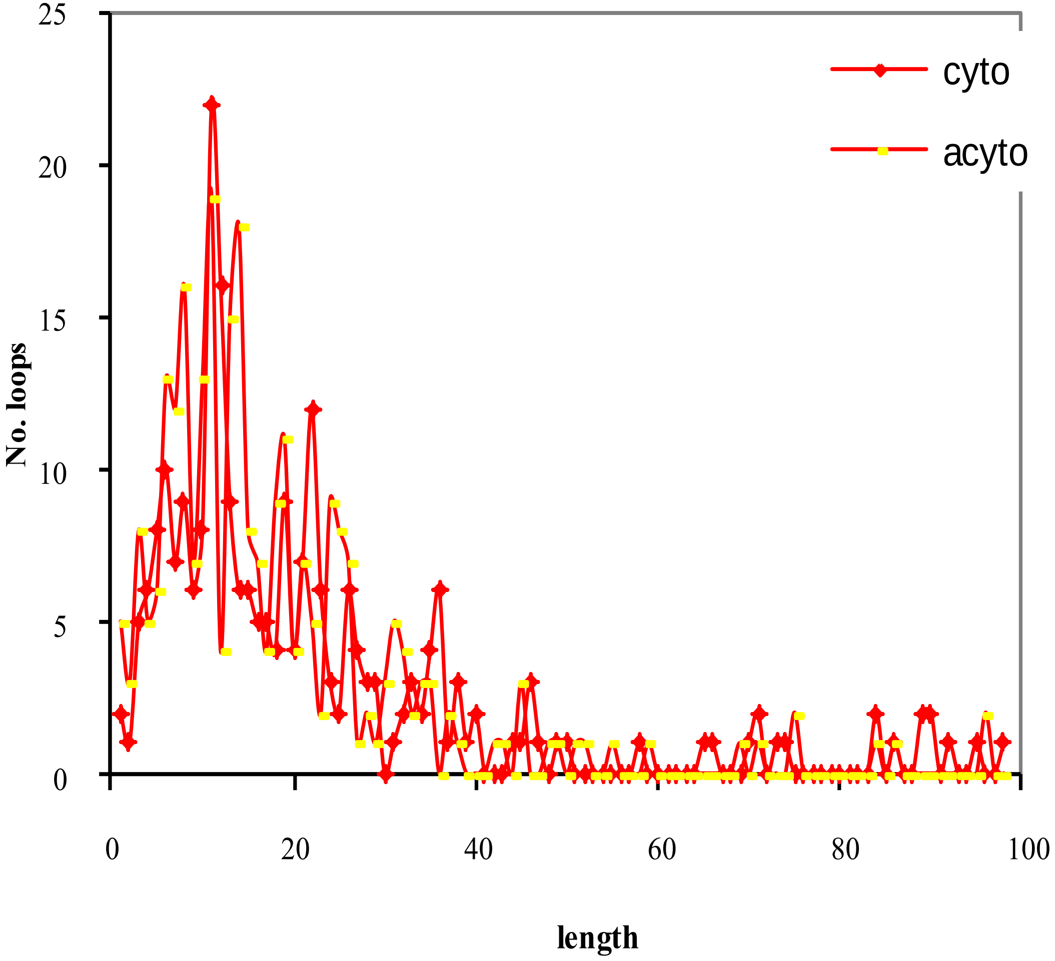


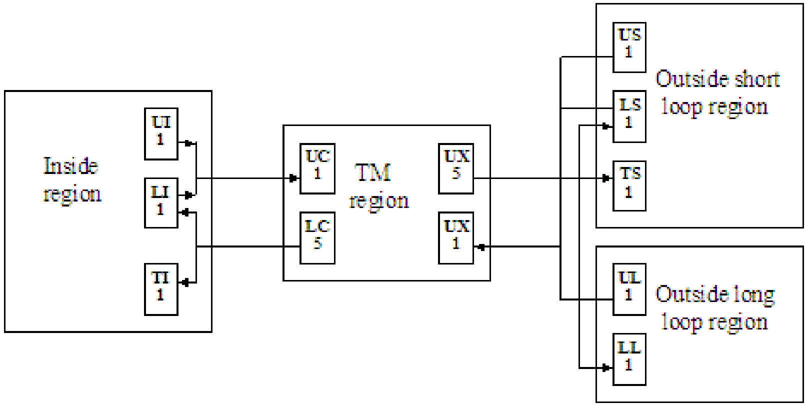
```
MFQLLAGVRMNSTGRPRAKIILLYALLIAFNIGAWLCALAAFRDHPVLLGT
iiiiiiiiiiiiiiiiiiiMMMMMMMMMMMMMMMMMMMMMooooooooooo
```

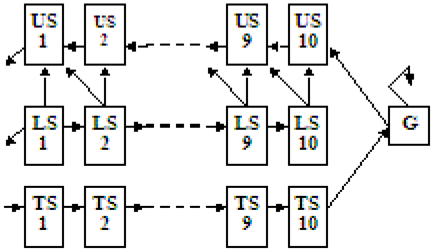
TMHMM   by Krogh, A. *et al* *JMB* **305**(2001)567-580

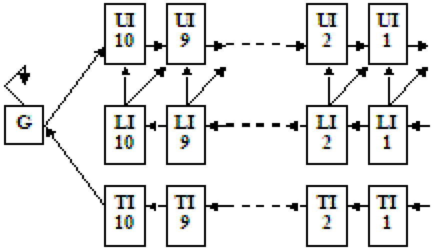


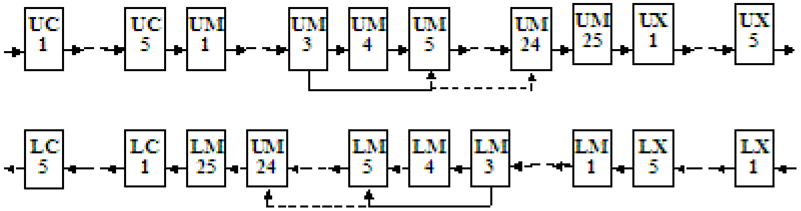
Accuracy of prediction for topology: 78%

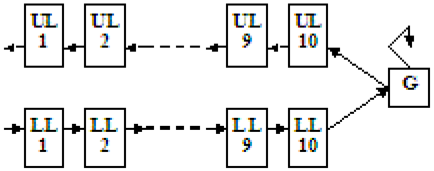

25
20
15
10
5
0
No. loops
0
20
40
60
80
100
length
cyto
acyto


Inside region
UI 1
LI 1
TI 1
UC 1
TM region
UX 5
LC 5
UX 1
US 1
Outside short loop region
LS 1
TS 1
UL 1
Outside long loop region
LL 1


US 1
US 2
US 9
US 10
LS 1
LS 2
LS 9
LS 10
G
TS 1
TS 2
TS 9
TS 10


UI 10
UI 9
UI 2
UI 1
G
LI 10
LI 9
LI 2
LI 1
TI 10
TI 9
TI 2
TI 1


UC 1
UC 5
UM 1
UM 3
UM 4
UM 5
UM 24
UM 25
UX 1
UX 5
LC 5
LC 1
LM 25
UM 24
LM 5
LM 4
LM 3
LM 1
LX 5
LX 1


UL 1
UL 2
UL 9
UL 10
G
LL 1
LL 2
LL 9
LL 10

Mod.	Reg.	Data set	Correct topology	Correct location	Sens-itivity	Speci-ficity
TMMOD 1	(a)	S-83	65 (78.3%)	67 (80.7%)	97.4%	97.4%
	(b)		51 (61.4%)	52 (62.7%)	71.3%	71.3%
	(c)		64 (77.1%)	65 (78.3%)	97.1%	97.1%
TMMOD 2	(a)	S-83	61 (73.5%)	65 (78.3%)	99.4%	97.4%
	(b)		54 (65.1%)	61 (73.5%)	93.8%	71.3%
	(c)		54 (65.1%)	66 (79.5%)	99.7%	97.1%
TMMOD 3	(a)	S-83	70 (84.3%)	71 (85.5%)	98.2%	97.4%
	(b)		64 (77.1%)	65 (78.3%)	95.3%	71.3%
	(c)		**74 (89.2%)**	**74 (89.2%)**	**99.1%**	**97.1%**
TMHMM		S-83	**64 (77.1%)**	**69 (83.1%)**	**96.2%**	**96.2%**
PHDtm		S-83	**(85.5%)**	**(88.0%)**	**98.8%**	**95.2%**
TMMOD 1	(a)	S-160	117 (73.1%)	128 (80.0%)	97.4%	97.0%
	(b)		92 (57.5%)	103 (64.4%)	77.4%	80.8%
	(c)		117 (73.1%)	126 (78.8%)	96.1%	96.7%
TMMOD 2	(a)	S-160	120 (75.0%)	132 (82.5%)	98.4%	97.2%
	(b)		97 (60.6%)	121 (75.6%)	97.7%	95.6%
	(c)		118 (73.8%)	135 (84.4%)	98.4%	97.2%
TMMOD 3	(a)	S-160	120 (75.0%)	133 (83.1%)	97.8%	97.6%
	(b)		110 (68.8%)	124 (77.5%)	94.5%	98.1%
	(c)		**135 (84.4%)**	**143 (89.4%)**	**98.3%**	**98.1%**
TMHMM		S-160	**123 (76.9%)**	**134 (83.8%)**	**97.1%**	**97.7%**