Hidden Markov Models

HIDDEN MARKOV MODELS

1. **Objectives**
2. **Markov chains**
3. **Hidden Markov Models**
4. **Key Tasks and Algorithms**
5. **Alignment**
6. **Profile Hidden Markov Models**
7. **Alternative Approaches**
8. **Packages**

1. Objectives

The key objective is to uncover the **latent** structure present within a biological sequence

- **GENOMICS -** structural and coding regions within DNA/RNA sequences
  - genes
  - regulatory regions
  - CpG islands
  - repeat regions

- **PROTEOMICS**
  - secondary and higher order structure
  - regions of specified function
  - alignments
  - conserved/homologous regions

Two common tasks in the analysis of biological sequences

- pairwise alignment
- multiple alignment

SIMILARITY

- characters in DNA/protein sequences $x = \{x_i\}, y = \{y_j\}$
- each character takes possible values $a, b, ...$
- two probabilistic models
  - Random model R : character a occurs independently with probability q_a - gives joint probability for x and y

$$P(x, y|R) = P(x|R) P(y|R) = \left\{\prod_i q_{x_i}\right\}\left\{\prod_j q_{y_i}\right\}$$

  - Match model M: character a occurs homologously with character b with probability p_{ab}

$$P(x, y|M) = \left\{\prod_i p_{x_i y_j}\right\}$$


```
(a)
HBA_HUMAN  GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKL
           G+ +VK+HGKKV  A+++++AH+D++ +++++LS+LH  KL
HBB_HUMAN  GNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKL

(b)
HBA_HUMAN  GSAQVKGHGKKVADALTNAVAHV---D--DMPNALSALSDLHAHKL
           ++ ++++H+ KV   + +A  ++           +L+ L+++H+ K
LGB2_LUPLU NNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATLKNLGSVHVSKG

(c)
HBA_HUMAN  GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSD----LHAHKL
           GS+ + G +   +D L  ++ H+ D+  A +AL D    ++AH+
F11G11.2   GSGYLVGDSLTFVDLL--VAQHTADLLAANAALLDEFPQFKAHQE
```

Figure 1: Sequence Alignments to human alpha globin HBA_HUMAN; (a) human beta globin HBB_HUMAN, (b) lupin leghaemoglobin LGB2_LUPLU, (c) nematode glutathione S-transferase F11G11.2.

- likelihood or odds ratio

$$LR = \frac{P(x, y|M)}{P(x, y|R)} = \frac{\left\{\prod_i p_{x_i y_j}\right\}}{\left\{\prod_i q_{x_i}\right\}\left\{\prod_j q_{y_i}\right\}} = \prod_i \frac{p_{x_i y_j}}{q_{x_i} q_{y_i}}$$

- on the log scale (usually log base 2)

$$S = \log LR = \sum_i s(x_i, y_i)$$

where

$$s(a, b) = \log \frac{p_{ab}}{q_a q_b}$$

This method is appropriate for two sequences of the same length; it can be extended by the inclusion of gaps, so that a gapped alignment can be obtained. Gaps of length g should be penalized by a **linear score**

$$\gamma(g) = -gd$$

or an **affine score**

$$\gamma(g) = -d - (g-1)e$$

where d is a penalty for having a gap at all, and e is the gap extension penalty.

The gap can be incorporated into the likelihood calculation by writing

$$\gamma(g) = \log f(g) \Longleftrightarrow f(g) = 2^{\gamma(g)}$$

and then incoporating $f(g)$ into the likelihood.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	**5**	−2	−1	−2	−1	−1	−1	0	−2	−1	−2	−1	−1	−3	−1	1	0	−3	−2	0
R	−2	**7**	−1	−2	−4	1	0	−3	0	−4	−3	3	−2	−3	−3	−1	−1	−3	−1	−3
N	−1	−1	**7**	2	−2	0	0	0	1	−3	−4	0	−2	−4	−2	1	0	−4	−2	−3
D	−2	−2	2	**8**	−4	0	2	−1	−1	−4	−4	−1	−4	−5	−1	0	−1	−5	−3	−4
C	−1	−4	−2	−4	**13**	−3	−3	−3	−3	−2	−2	−3	−2	−2	−4	−1	−1	−5	−3	−1
Q	−1	1	0	0	−3	**7**	2	−2	1	−3	−2	2	0	−4	−1	0	−1	−1	−1	−3
E	−1	0	0	2	−3	2	**6**	−3	0	−4	−3	1	−2	−3	−1	−1	−1	−3	−2	−3
G	0	−3	0	−1	−3	−2	−3	**8**	−2	−4	−4	−2	−3	−4	−2	0	−2	−3	−3	−4
H	−2	0	1	−1	−3	1	0	−2	**10**	−4	−3	0	−1	−1	−2	−1	−2	−3	2	−4
I	−1	−4	−3	−4	−2	−3	−4	−4	−4	**5**	2	−3	2	0	−3	−3	−1	−3	−1	4
L	−2	−3	−4	−4	−2	−2	−3	−4	−3	2	**5**	−3	3	1	−4	−3	−1	−2	−1	1
K	−1	3	0	−1	−3	2	1	−2	0	−3	−3	**6**	−2	−4	−1	0	−1	−3	−2	−3
M	−1	−2	−2	−4	−2	0	−2	−3	−1	2	3	−2	**7**	0	−3	−2	−1	−1	0	1
F	−3	−3	−4	−5	−2	−4	−3	−4	−1	0	1	−4	0	**8**	−4	−3	−2	1	4	−1
P	−1	−3	−2	−1	−4	−1	−1	−2	−2	−3	−4	−1	−3	−4	**10**	−1	−1	−4	−3	−3
S	1	−1	1	0	−1	0	−1	0	−1	−3	−3	0	−2	−3	−1	**5**	2	−4	−2	−2
T	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	2	**5**	−3	−2	0
W	−3	−3	−4	−5	−5	−1	−3	−3	−3	−3	−2	−3	−1	1	−4	−4	−3	**15**	2	−3
Y	−2	−1	−2	−3	−3	−1	−2	−3	2	−1	−1	−2	0	4	−3	−2	−2	2	**8**	−1
V	0	−3	−3	−4	−1	−3	−3	−4	−4	4	1	−3	1	−1	−3	−2	0	−3	−1	**5**

Figure 2: BLOSUM 50 protein scoring matrix

ALIGNMENT ALGORITHMS

There are several alignment algorithms that produce the optimal alignment between two sequences:

- Global Alignment (**NEEDLEMAN-WUNSCH**)
- Local Alignment (**SMITH-WATERMAN**)
- Gapped Alignment
- Repeats
- Overlaps

	H	E	A	G	A	W	G	H	E	E
P	−2	−1	−1	−2	−1	−4	−2	−2	−1	−1
A	−2	−1	**5**	0	**5**	−3	0	−2	−1	−1
W	−3	−3	−3	−3	−3	**15**	−3	−3	−3	−3
H	**10**	0	−2	−2	−2	−3	−2	**10**	0	0
E	0	**6**	−1	−3	−1	−3	−3	0	**6**	**6**
A	−2	−1	**5**	0	**5**	−3	0	−2	−1	−1
E	0	**6**	−1	−3	−1	−3	−3	0	**6**	**6**

Figure 3: Two sequences $HEAGAWGHEE$ and $PAWHEAE$

These two sequences are of different lengths, and hence the introduction of gaps is necessary.

The global alignment is constructed recursively from left to right across the sequence. The steps are as follows:

1. Construct a matrix F of zeros of size $(n+1, m+1)$: F will store the best aligment, that is $F(i,j)$ contains the best alignment score for the initial sequences $x_{12\ldots i}$ and $y_{12\ldots j}$
2. Define $F(i,j)$ recursively in terms of $F(i-1,j-1)$, $F(i-1,j)$ and $F(i,j-1)$

   $F(i,j) = \max\{F(i-1,j-1) + s(x_i, y_j), F(i-1,j) - d, F(i,j-1) - d\}$

   - $F(i-1,j-1) + s(x_i, y_j)$ is the score for aligning x_i and y_j
   - $F(i-1,j) - d$ is the score for aligning x_i to a gap
   - $F(i,j-1) - d$ is the score for aligning y_j to a gap
3. Define $F(0,0) = 0, F(i,0) = -id, F(0,j) = -jd$

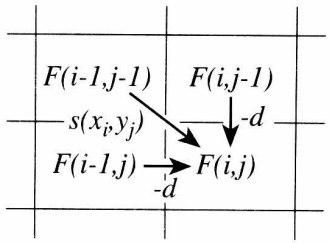

Figure 4: Recursive calculation of the F matrix

We now seek an optimal path through the matrix F

The optimal path through the matrix F is found by tracing back up through the matrix from the bottom right hand corner.

This algorithm can be extended to allow for

- optimal local alignment (SMith-Waterman)
- overlaps
- gaps

The only aspect that changes is the nature of the recursion

$$F(i,j) = \max\{F(i-1,j-1) + s(x_i, y_j), F(i-1,j) - d, F(i,j-1) - d\}$$

		H	E	A	G	A	W	G	H	E	E
	0	**-8**	**-16**	-24	-32	-40	-48	-56	-64	-72	-80
P	-8	-2	-9	**-17**	**-25**	-33	-42	-49	-57	-65	-73
A	-16	-10	-3	-4	-12	**-20**	-28	-36	-44	-52	-60
W	-24	-18	-11	-6	-7	-15	**-5**	**-13**	-21	-29	-37
H	-32	-14	-18	-13	-8	-9	-13	-7	**-3**	-11	-19
E	-40	-22	-8	-16	-16	-9	-12	-15	-7	**3**	-5
A	-48	-30	-16	-3	-11	-11	-12	-12	-15	**-5**	2
E	-56	-38	-24	-11	-6	-12	-14	-15	-12	-9	**1**

```
HEAGAWGHE-E
--P-AW-HEAE
```

Figure 5: The optimal global alignment.

With an affine gap score three matrices/recurrence relations are needed:

- For alignment of x_i with y_j, $M(i,j)$

$$M(i,j) = \max \begin{cases} M(i-1,j-1) + s(x_i, y_j) \\ I_x(i-1,j-1) + s(x_i, y_j) \\ I_y(i-1,j-1) + s(x_i, y_j) \end{cases}$$

- For alignment of x_i with a gap, $I_x(i,j)$

$$I_x(i,j) = \max\{M(i-1,j) - d, I_x(i-1,j) - e\}$$

- For alignment of y_j with a gap, $I_y(i,j)$

$$I_y(i,j) = \max\{M(i,j-1) - d, I_y(i,j-1) - e\}$$

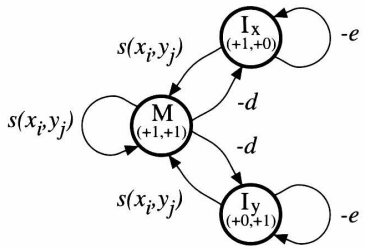

Figure 6: A diagram of the gapped alignment algorithm

This "state" diagram indicates how the alignment algorithm works.

2. Markov chains

Markov chains are a special type of *discrete* valued, *discrete* time stochastic. Consider

- a sequence of discrete random variables $\{X_t\}$ indexed in discrete time by $t = 1, 2, 3, ...,$
- and each having a range of possible values (or **states**) $\mathbb{X} = \{1, 2, 3, ..., S\}$ say (with S finite).
- joint distribution of the $\{X_1, X_2, X_3, ...\}$ is specified via the one-step ahead conditional probabilities

  $$\Pr[X_{t+1} = x_{t+1} | X_1 = x_1, ..., X_t = x_t] = \Pr[X_{t+1} = x_{t+1} | X_t = x_t] = p_{ij}$$

  which specifies $\Pr(\text{State } i \text{ at time } t \rightarrow \text{State } j \text{ at time } t+1)$ for $i, j \in \{1, 2, 3, ..., S\}$, say, that **does not depend on** t.

The probabilistic specification can be encapsulated in the $S \times S$ matrix

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1S} \\ p_{21} & p_{22} & \cdots & p_{2S} \\ \vdots & \vdots & \ddots & \vdots \\ p_{S1} & p_{S2} & \cdots & p_{SS} \end{bmatrix}$$

which is called the **transition matrix** where the element in row i and column j defines the probability of moving from state i to state j.

The sequence of random variables described by the matrix P form a **Markov Chain**. To complete the specification, a probability specification for the initial state random variable X_1 is required

$$\pi^{(1)} = \left(\pi_1^{(1)}, \pi_2^{(1)}, ...\pi_S^{(1)}\right)$$

To compute the (marginal) probability of random variable X_t taking the value i, we can use matrix algebra and an iterative calculation as follows; let $\pi^{(t)} = \left(\pi_1^{(t)}, \pi_2^{(t)}, ...\pi_S^{(t)}\right)$ denote the probability distribution of X_t, then

$$\pi_j^{(t)} = \sum_{i=1}^{S} p_{ij} \pi_i^{(t-1)}$$

or in matrix form

$$\pi^{(t)} = \pi^{(t-1)} P$$

Using this definition recursively, we have

$$\pi^{(t)} = \pi^{(t-1)} P = \pi^{(t-2)} P^2 = \pi^{(t-3)} P^3 = ... = \pi^{(2)} P^{t-2} = \pi^{(1)} P^{t-1}$$

which gives a mechanism for computing the marginal probability after t steps.

3. Hidden Markov models

Suppose that a nucleotide position can be classified as being in a **region** of one of a number, H. Then, we might assume that, within a homogeneous region labelled by h the observed nucleotide sequence follows a Markov chain with transition matrix P_h

To complete the specification, we assume that there is a sequence in parallel to the nucleotide sequence, comprising region label random variables $H_1, H_2, ...$ which itself follows a Markov chain governed by transition matrix P_θ where for $i, j \in \{1, 2, 3, ..., H\}$

$$\mathrm{P}\left[H_{t+1} = j | H_t = i\right] = \Pr\left(\text{Region type } i \text{ at time } t \rightarrow \text{Region type } j \text{ at time } t+1\right)$$

The key difference is that the $\{H_t\}$ sequence is **unobserved** or **latent**.

EXAMPLE CpG Islands (Durbin et al. (1998))
CG or CpG dinucleotides are common in certain parts of the human genome.

Regions that are CpG rich are called CpG islands, and are typically up to 1000 bases long.

It is of biological interest to determine the locations of CpG islands, as these regions are often associated with, for example, promoter sites.

$$\cdots CGCGTACGCGCGTGAC \cdots$$

From the observed sequence alone, it is not possible to deduce whether a given CG pair is part of a CpG island, or is just a normal dinucleotide pair.

For the CpG island example, we can use an eight state hidden Markov model (HMM)

- A_+, C_+, G_+, T_+ are the states corresponding to nucleotides within a CpG region
- A_-, C_-, G_-, T_- are the states corresponding to nucleotides outside of a CpG region
- we can augment these eight states by two further states for completeness
  - a **begin** state
  - an **end** state

  These two states are "silent" as they do not contribute a nucleotide to the observed nucleotide sequence, but are useful for introducing gaps into the sequence.

As described above, some parameters control the nature of the Markov model. Let $h = (h_1, h_2, ...)$ denote the **hidden** state sequence, and $x = (x_1, x_2, ...)$ denote the **observed** nucleotide sequence

- **transition probabilities**

$$a_{kl} = \Pr\left(h_i = l | h_{i-1} = k\right)$$

- **emission probabilities**

$$e_k\left(b\right) = \Pr\left(x_i = b | h_i = k\right)$$

In the CpG island case, the emission probabilities are all either 1 or zero, for example

$$\Pr\left(x_i = A | h_i = A_+\right) = 1 \qquad \Pr\left(x_i = C | h_i = A_+\right) = 0$$

and so on.

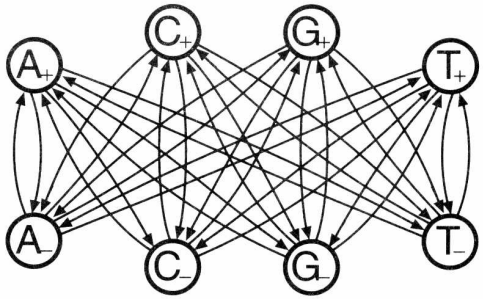

Figure 7: State model for CpG island HMM

Estimates of both sets of parameters can be obtained (using maximum likelihood) from training data. For the CpG example, let

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

denote the transition probabilities within a CpG island. Estimates can be derived from established CpG islands

+	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

Need also to assess probabilities of transitions between region types, for example $A_+ \rightarrow C_-$.

To assess whether a sequence is more likely to be a segment from a CpG island, we may use a likelihood ratio-type test based on

$$S(x) = \log \frac{P\left(x|\text{ CpG island}\right)}{P\left(x|\text{ not CpG island }\right)} = \sum_{i=1}^{L} \log \frac{a_{x_{i-1}x_i}^{+}}{a_{x_{i-1}x_i}^{-}} = \sum_{i=1}^{L} \beta_{x_{i-1}x_i}$$

where

β	A	C	G	T
A	-0.740	0.419	0.580	-0.803
C	-0.913	0.302	1.812	-0.685
G	-0.624	0.461	0.331	-0.730
T	-1.169	0.573	0.393	-0.679

4. Key Tasks and Algorithms

TASK 1: **The most probable state path: The Viterbi Algorithm:**

$$\begin{aligned} \text{We seek } \; h^* &= \arg\max_h P\left(x, h\right) \\ &= \arg\max_h P\left(h|x\right) \end{aligned}$$

That is, we seek the most probable path or sequence of states in light of the data

1.
   - $P\left(x, h\right)$ is the joint probability or joint likelihood
   - $P\left(h|x\right)$ is the posterior probability

(i) Initialize vector $v_0(0) = 1, v_k(0) = 0$ for $k > 0$

(ii) Recursion: for $i = 1, ..., L$

$$\begin{aligned} v_l(i) &= e_l(x_i) \max_k (v_k(i-1) a_{kl}) \\ point_i(l) &= \arg\max_k (v_k(i-1) a_{kl}) \end{aligned}$$

(iii) Final recursion:

$$\begin{aligned} P(x, h^*) &= \max_k (v_k(L) a_{k0}) \\ h_L^* &= \arg\max_k (v_k(L) a_{k0}) \end{aligned}$$

(iv) Trace back the definition of the optimal path

$$h_{i-1}^* = point_i(h_i^*)$$

TASK 2: **The probability of an observed sequence: The Forward Algorithm:**

$$\text{We seek } P(x) = \sum_h P(x, h)$$

However, the direct computation of this sum is not possible, so a recursive calculation is needed.

(i) Initialize vector $f_0(0) = 1, f_k(0) = 0$ for $k > 0$

(ii) Recursion: for $i = 1, ..., L$

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

(iii) Final recursion:

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

TASK 3: **The posterior probabilities: The Backward Algorithm:**

$$\text{We seek } P(h_i = k|x) = \frac{P(h_i = k, x)}{P(x)} = \frac{P(x_1, ..., x_i, h_i = k) P(x_{i+1}, ..., x_L | h_i = k)}{P(x)}$$

However, the direct computation of this sum is not possible, so a recursive calculation is needed.

(i) Initialize element $b_k(L) = a_{k0}$

(ii) Backwards recursion: for $i = L - 1, ..., 1$

$$b_k(i) = \sum_k a_{kl} e_l(x_{i+1}) b_l(i+1)$$

(iii) Final recursion:

$$P(x) = \sum_k a_{k0} e_l(x_1) b_l(1)$$

We then have

$$P(h_i = k, x) = f_k(i) b_k(i)$$

as the forward and backward probabilities, $f_k(i)$ and $b_k(i)$, found by the Forward and Backward Algorithms, are precisely

$$f_k(i) = P(x_1, ..., x_i, h_i = k)$$

$$b_k(i) = P(x_{i+1}, ..., x_L | h_i = k)$$

and hence

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

TASK 4: **Parameter Estimation: The Baum-Welch Algorithm:** The algorithms above are all espressed in terms of known transition and emission probabilities. These probabilities can be pre-specified, estimated from training sequences, or estimated during the state sequence reconstruction.

If the state sequence is **known**, let A_{kl} be the number of transitions from state k to state l, and let $E_k(b)$ be the number of times character b is emitted from state k. Then we may estimate the transition and emission probabilities as follows

$$\widehat{a}_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \qquad \widehat{e}_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')} \tag{1}$$

If the state sequence is **not known**, we must reconstruct the state sequence and perform estimation concurrently. This is achieved using the **Baum-Welch** algorithm, which is a special case of a more general statistical algorithm called the **Expectation-Maximization** (or **EM)** algorithm.

The Baum-Welch algorithm calculates A_{kl} and $E_k(b)$ as the expected number of transitions and emissions, using the forward and backward algorithms. The probability of a transition $k \to l$ at position i is

$$P(h_i = k, h_{i+1} = l|x) = \frac{f_k(i)\, a_{kl} e_l(x_{i+1})\, b_l(i+1)}{P(x)}$$

and thus the expected number of times this transition will be observed is

$$A_{kl} = \sum_j \frac{1}{P(x_j)} \left\{ \sum_i f_k^j(i)\, a_{kl} e_l\left(x_{i+1}^j\right) b_l^j(i+1) \right\}$$

In this formula

- x_j is the j^{th} training sequence
- $f_k^j(i)$ is the forward probability from the forward algorithm implemented for sequence j
- $b_l^j(i)$ is the backward probability from the backward algorithm implemented for sequence j

Similarly,

$$E_k(b) = \sum_j \frac{1}{P(x_j)} \left\{ \sum_{i \in \mathcal{B}_j} f_k^j(i)\, b_l^j(i) \right\}$$

where $\mathcal{B}_j$ is the set of indices of locations in training sequence j whhere character b is emitted.

Hence we have the following algorithm:

(i) Initialize parameter values to some arbitrary values

(ii) Recursion:

(a) Set $A_{kl}^{(0)}$, $E_k^{(0)}(b)$ to be some small "prior" count value

(b) For each sequence $j = 1, ..., n$

I Calculate $f_k^j(i)$ using the Forward Algorithm

II Calculate $b_k^j(i)$ using the Backward Algorithm

III Set

$$A_{kl}^{(j)} = A_{kl}^{(j-1)} + \frac{1}{P(x_j)} \left\{ \sum_i f_k^j(i)\, a_{kl} e_l\left(x_{i+1}^j\right) b_l^j(i+1) \right\}$$

and

$$E_k^{(j)}(b) = E_k^{(j-1)}(b) + \frac{1}{P(x_j)} \left\{ \sum_{i \in \mathcal{B}_j} f_k^j(i)\, b_l^j(i) \right\}$$

(c) Estimate a_{kl} and $e_k(b)$ using (1), and the log likelihood $P(x)$

(iii) Terminate the recursion if the computed $P(x)$ value compared to the previous iteration is virtually identical.

This algorithm is reasonably straightforward to implement given the previous algorithms.

5. More Complex State Structures

- **Silent States** These are states that do not emit characters, but can be used as placeholders in a HMM state sequence; this is necessary to allo the modelling of insertions and deletions. In the state space model below, all positions in the sequence (indicated by squares) intercommunicate with all positions later in the chain.

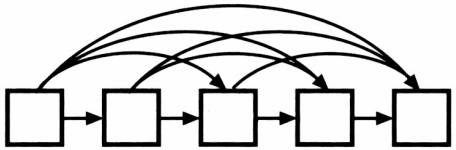

Figure 8: Chain without Silent States

An alternative representation uses silent states (circles):

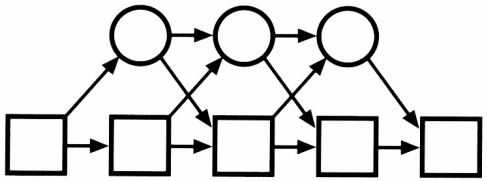

Figure 9: Chain withSilent States

This model is more restricted than the fully connected (backward and forward) model, having many fewer parameters, but is therefore more readily estimable from training data. This model can be used for arbitrary deletions in a biological sequence.

- **Higher-order Markov chains:** It is plausible biologically, in many cases, to allow for dependence between di- and tri-nuclotides in DNA sequences, or similar extended structures in protein structures.
- **Inhomogeneous Markov Chains:** Such models allow for structural changes in the latent state sequence, which might be necessary or advantageous in the analysis of long sequences
- **Variability in emission probabilities**
- **Non-global alignments**

6. Alignment Using HMMs

The ideas introduced for the alignment of sequences earlier can be reformulated within a HMM framework. We require three states

- M match state
- X insertion into sequence x
- Y insertion into sequence y

The deterministic structure introduced previously be rexpressed using the new HMM notation. Below, the left-hand figure represents the deterministic optimal alignment, whereas the right-hand figure represents the HMM model for alignment

The advantage of using a probabilistic approach is that instead of merely obtaining the optimal alignment, and an alignment score, the following quantities are also obtained

- posterior probability of alignment
- pointwise alignment uncertainty measures
- assessment of suboptimal alignments
- possibility of incorporation into subsequent analysis (homology, protein classification) in a coherent fashion

These advantages come at the price of extra computational burden.

The pairwise alignment can readily be extended to more complex situations

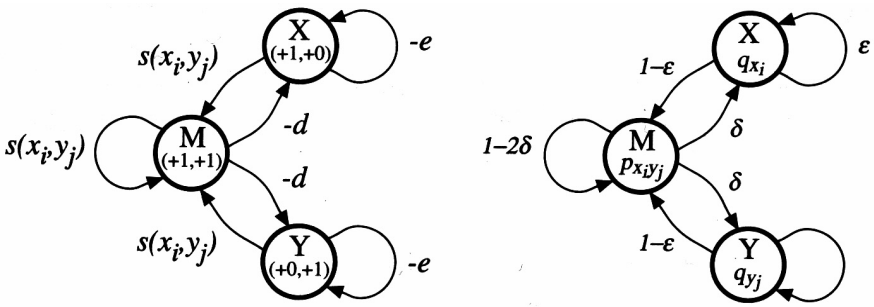

Figure 10: Chain for alignment

7. Multiple Alignment : Profile HMMs

The final extension to the ordinary HMM modelling strategy permits multiple alignment in a fully probabilistic framework.

Typically, we will have a number of apparently homologous sequences for which we want an optimal alignment; below we have an example from proteomics

- seven globins to be aligned
- secondary structure (helix presence) also of interest.
- consensus sequence recorded

```
Helix                AAAAAAAAAAAAAAA   BBBBBBBBBBBBBBBBCCCCCCCCCCC
HBA_HUMAN  ---------VLSPADKTNVKAAWGKVGA--HAGEYGAEALERMFLSFPTTKTYFPHF
HBB_HUMAN  --------VHLTPEEKSAVTALWGKV---NVDEVGGEALGRLLVVYPWTQRFFESF
MYG_PHYCA  ---------VLSEGEWQLVLHVWAKVEA--DVAGHGQDILIRLFKSHPETLEKFDRF
GLB3_CHITP ----------LSADQISTVQASFDKVKG------DPVGILYAVFKADPSIMAKFTQF
GLB5_PETMA PIVDTGSVAPLSAAEKTKIRSAWAPVYS--TYETSGVDILVKFFTSTPAAQEFFPKF
LGB2_LUPLU --------GALTESQAALVKSSWEEFNA--NIPKHTHRFFILVLEIAPAAKDLFS-F
GLB1_GLYDI ---------GLSAAQRQVIAATWKDIAGADNGAGVGKDCLIKFLSAHPQMAAVFG-F
Consensus            Ls....  v a W kv .  .    g . L.. f . P .   F  F

Helix         DDDDDDDEEEEEEEEEEEEEEEEEE          FFFFFFFFFFF
HBA_HUMAN  -DLS-----HGSAQVKGHGKKVADALTNAVAHV---D--DMPNALSALSDLHAHKL-
HBB_HUMAN  GDLSTPDAVMGNPKVKAHGKKVLGAFSDGLAHL---D--NLKGTFATLSELHCDKL-
MYG_PHYCA  KHLKTEAEMKASEDLKKHGVTVLTALGAILKK----K-GHHEAELKPLAQSHATKH-
GLB3_CHITP AG-KDLESIKGTAPFETHANRIVGFFSKIIGEL--P---NIEADVNTFVASHKPRG-
GLB5_PETMA KGLTTADQLKKSADVRWHAERIINAVNDAVASM--DDTEKMSMKLRDLSGKHAKSF-
LGB2_LUPLU LK-GTSEVPQNNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATLKNLGSVHVSKG-
GLB1_GLYDI SG----AS---DPGVAALGAKVLAQIGVAVSHL--GDEGKMVAQMKAVGVRHKGYGN
Consensus   .  t    .. . .v..Hg kv. a   a...l   d   . a l. l   H  .

Helix       FFGGGGGGGGGGGGGGGGGG     HHHHHHHHHHHHHHHHHHHHHHHHH
HBA_HUMAN  -RVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR------
HBB_HUMAN  -HVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH------
MYG_PHYCA  -KIPIKYLEFISEAIIHVLHSRHPGDFGADAQGAMNKALELFRKDIAAKYKELGYQG
GLB3_CHITP --VTHDQLNNFRAGFVSYMKAHT--DFA-GAEAAWGATLDTFFGMIFSKM-------
GLB5_PETMA -QVDPQYFKVLAAVIADTVAAG---------DAGFEKLMSMICILLRSAY-------
LGB2_LUPLU --VADAHFPVVKEAILKTIKEVVGAKWSEELNSAWTIAYDELAIVIKKEMNDAA---
GLB1_GLYDI KHIKAQYFEPLGASLLSAMEHRIGGKMNAAAKDAWAAAYADISGALISGLQS-----
Consensus    v.   f  l . ..  ....   f   . aa. k. .    l sky
```

Figure 11: Profile Hidden Markov Model

The only necesary extension is the idea of a **delete** state;All other aspects

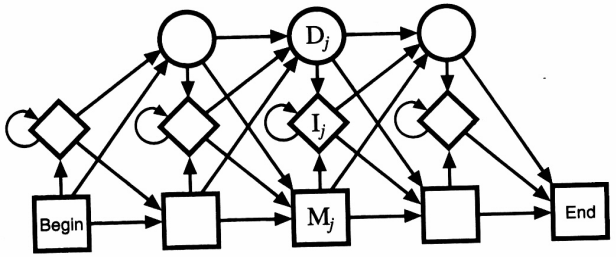

Figure 12: Chain with delete states

of the model proceed as before: however, as the models become more sophisticated, as the structures to be modelled become more complex, the amount of training data needed becomes large.

8. Alternative aproaches

Two alternative or complementary approaches to inference are worthy of mention:

1. **Markov chain Monte Carlo**: a general inference method that is used to make Bayesian (rather than purely likelihood-based) inference about the parameters of interest. An iterative (indeed Markov chain) scheme is used to draw samples from the posterior distributions of the state sequence, transition and emission probabilities. This approach retains the probabilistic structure of the HMM, but performs the inference in a different manner.

2. **Neural Networks:** Especially useful in protein structure prediction, the HMM model is replaced by a non-linear regression model,

9. Packages

- HMMER
- SAM: Sequence Alignment and Modeling
- TMHMM (transmembrane proteins)

Key Reference: *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, R. Durbin, S. Eddy, A Krogh, G. Mitchison, Cambridge University Press, Cambridge, 1998