HMM for sequence alignment: profile HMM

Pair HMM

HMM for pairwise sequence alignment, which incorporates affine gap scores.

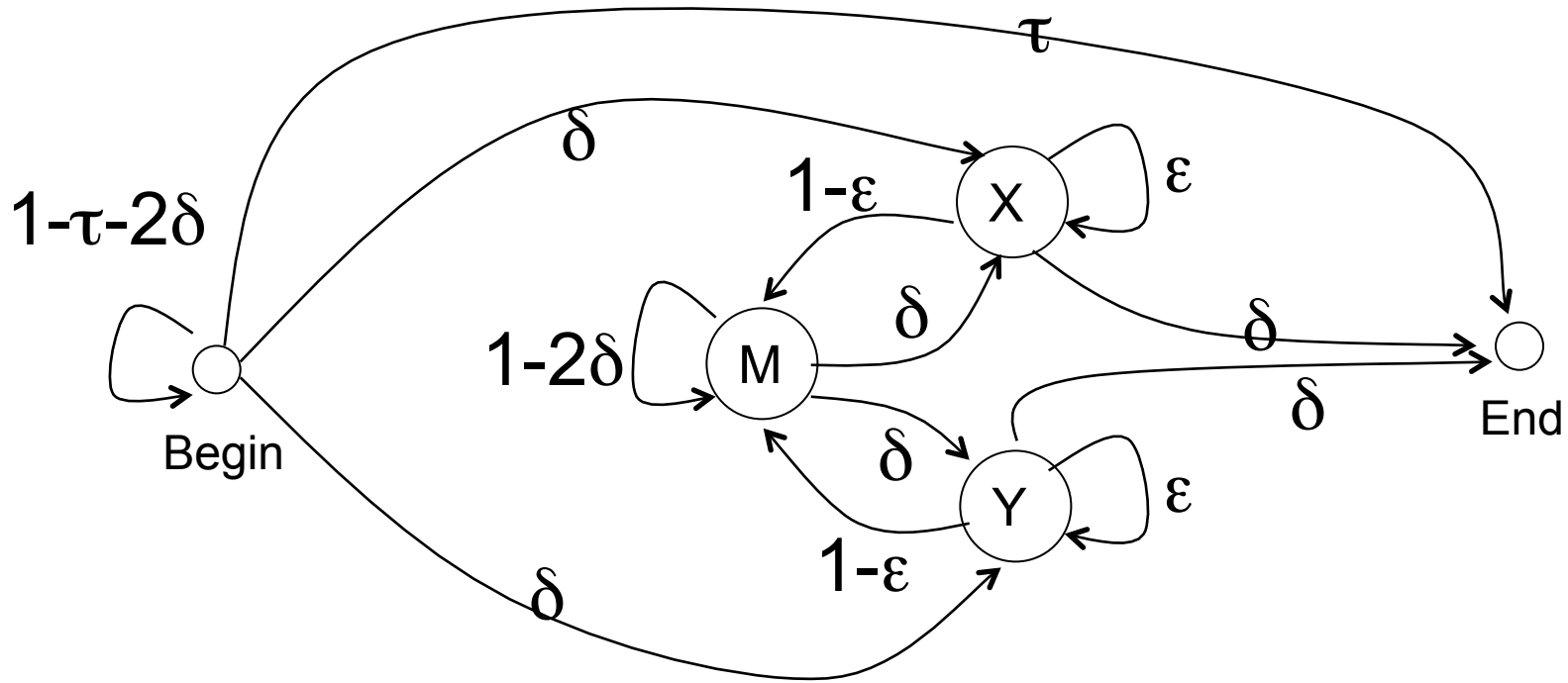
"Hidden" States

- Match (M)
- Insertion in *x* (X)
- insertion in *y* (Y)

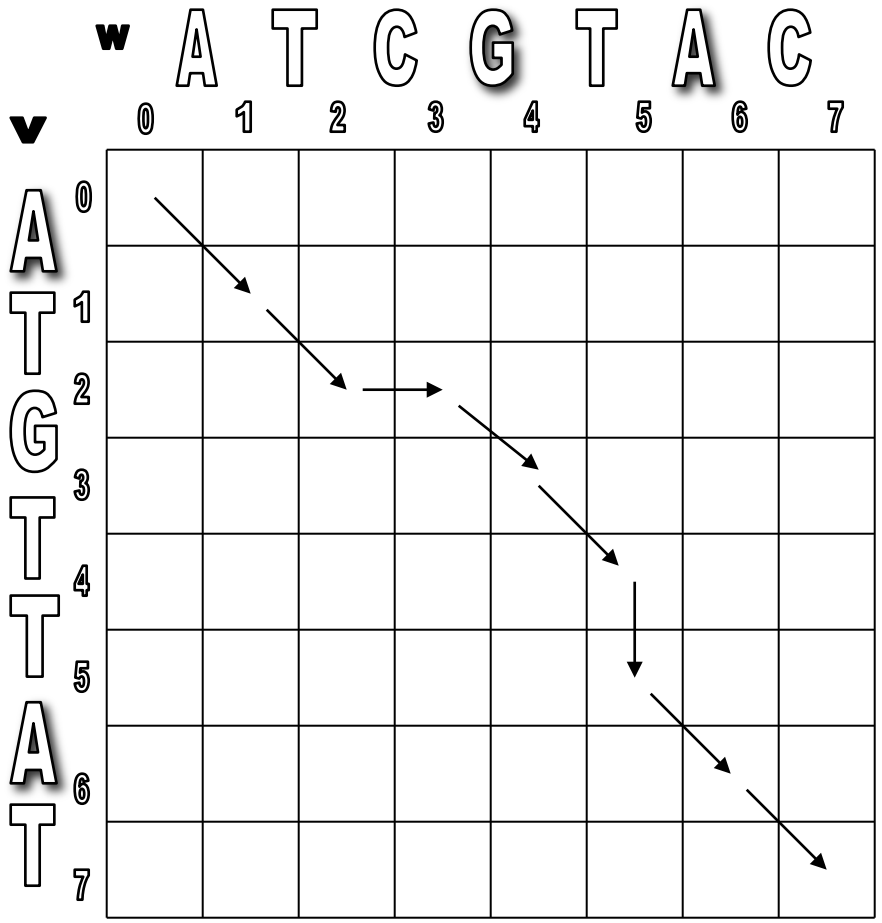
Observation Symbols

- Match (M): {(*a*,*b*)| *a*,*b* in ∑ }.
- Insertion in *x* (X): {(*a*,-)| *a* in ∑ }.
- Insertion in *y* (Y): {(-,*a*)| *a* in ∑ }.

Pair HMMs



Alignment: a path → a hidden state sequence



A T – G T T A T
A T C G T – A C

M M Y M M X M M

Multiple sequence alignment (Globin family)

Profile model (PSSM)

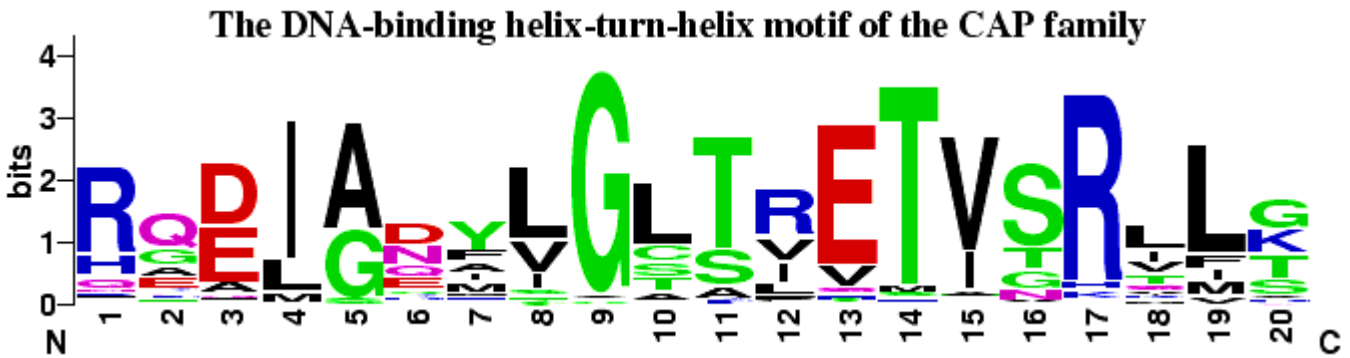
- A natural probabilistic model for a conserved region would be to specify independent probabilities $e_i(a)$ of observing nucleotide (amino acid) *a* in position *i*
- The probability of a new sequence x according to this model is

$$P(x \mid M) = \prod_{i=1}^{L} e_i(x_i)$$

Profile / PSSM

- DNA / proteins Segments of the same length L;
- Often represented as Positional frequency matrix;

```
LTMTRGDIGNYLGLTVETISRLLGRFQKSGML
LTMTRGDIGNYLGLTIETISRLLGRFQKSGMI
LTMTRGDIGNYLGLTVETISRLLGRFQKSEIL
LTMTRGDIGNYLGLTVETISRLLGRLQKMGIL
LAMSRNEIGNYLGLAVETVSRVFSRFQQNELI
LAMSRNEIGNYLGLAVETVSRVFTRFQQNGLI
LPMSRNEIGNYLGLAVETVSRVFTRFQQNGLL
VRMSREEIGNYLGLTLETVSRLFSRFGREGLI
LRMSREEIGSYLGLKLETVSRTLSKFHQEGLI
LPMCRRDIGDYLGLTLETVSRALSQLHTQGIL
LPMSRRDIADYLGLTVETVSRAVSQLHTDGVL
LPMSRQDIADYLGLTIETVSRTFTKLERHGAI
```

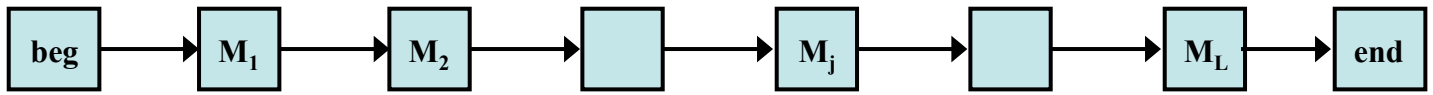


Searching profiles: inference

- Give a sequence S of length L, compute the likelihood ratio of being generated from this profile vs. from background model:
  - R(S|P)= $\prod_{i=1}^{L} \frac{e_i(x_i)}{q_{x_i}}$
  - Searching motifs in a sequence: sliding window approach

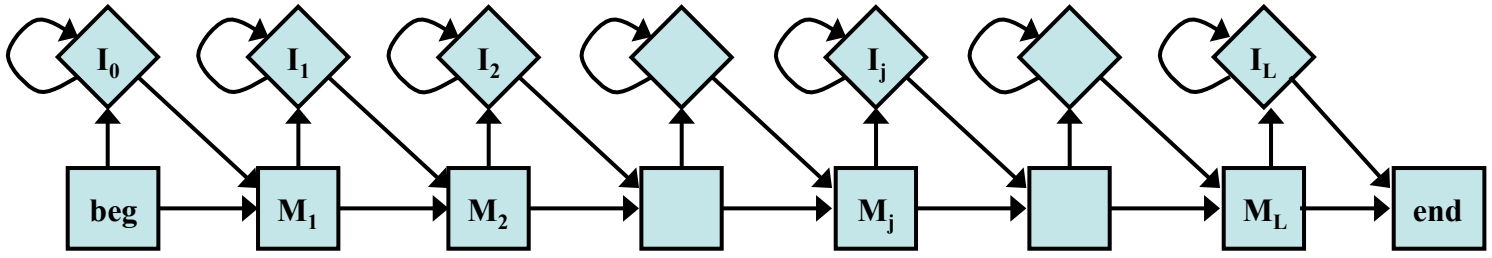
Match states for profile HMMs

- Match states
  - Emission probabilities $e_{M_i}(a)$



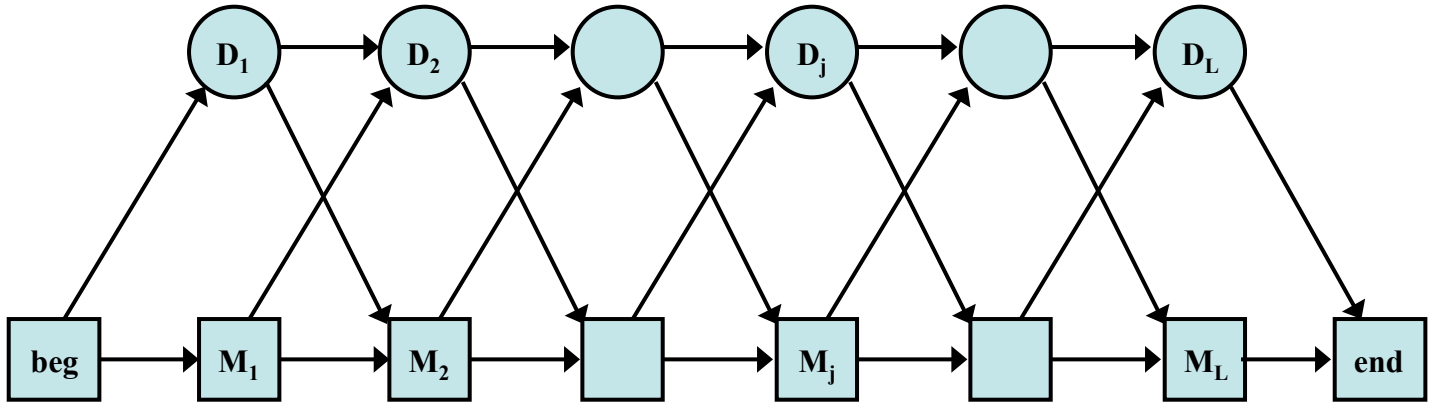
Components of profile HMMs

- Insert states $e_{\mathrm{I}_i}(a)$
  - Emission prob.
    - Can be the background distribution q_a.
  - Transition prob.
    - M_i to I_i, I_i to itself, I_i to M_{i+1}
  - Log-score for a gap of length k (not including the log-score from emission) $\log a_{\mathrm{M}_j\mathrm{I}_j} + \log a_{\mathrm{I}_j\mathrm{M}_j+1} + (k-1)\log a_{\mathrm{I}_j\mathrm{I}_j}$

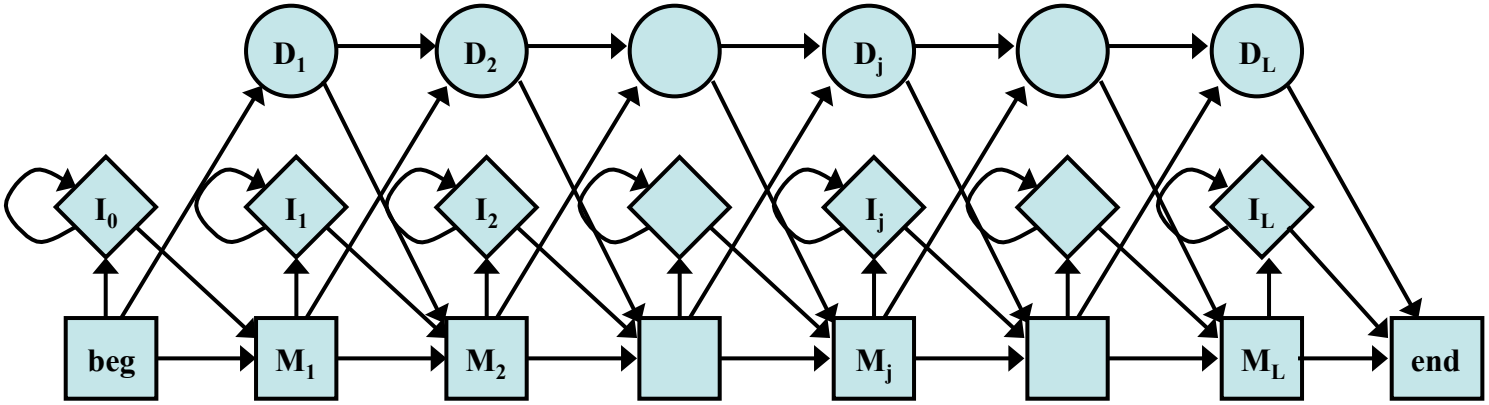


Components of profile HMMs

- Delete states
  - No emission prob.
  - Cost of a deletion
    - M_i to D_{i+1}, D_i to D_{i+1}, D_i to M_{i+1}
    - Each D_i to D_{i+1} might be different for different i



Full structure of profile HMMs



This is the structure implemented in Hmmer, slightly different from the structure described in the textbook: there is no transition allowed from D_j to I_j or from I_j to D_{j+1}. As a result, the recursive equation for Viterbi algorithm is different from the one described in the book too.

Deriving HMMs from multiple alignments

- Key idea behind profile HMMs
  - Model representing the consensus for the alignment of sequence from the same family
  - Not the sequence of any particular member

```
HBA_HUMAN    ...VGA--HAGEY...
HBB_HUMAN    ...V----NVDEV...
MYG_PHYCA    ...VEA--DVAGH...
GLB3_CHITP   ...VKG------D...
GLB5_PETMA   ...VYS--TYETS...
LGB2_LUPLU   ...FNA--NIPKH...
GLB1_GLYDI   ...IAGADNGAGV...
                  ***  *****
```

Deriving HMMs from multiple alignments

- Basic profile HMM parameterization
  - Aim: making the higher probability for sequences from the family
- Parameters
  - the transition and emission probabilities: trivial if many of independent alignment sequences are given.

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \qquad e_k(a) = \frac{E_k(a)}{\sum_{a'} E_k(a')}$$

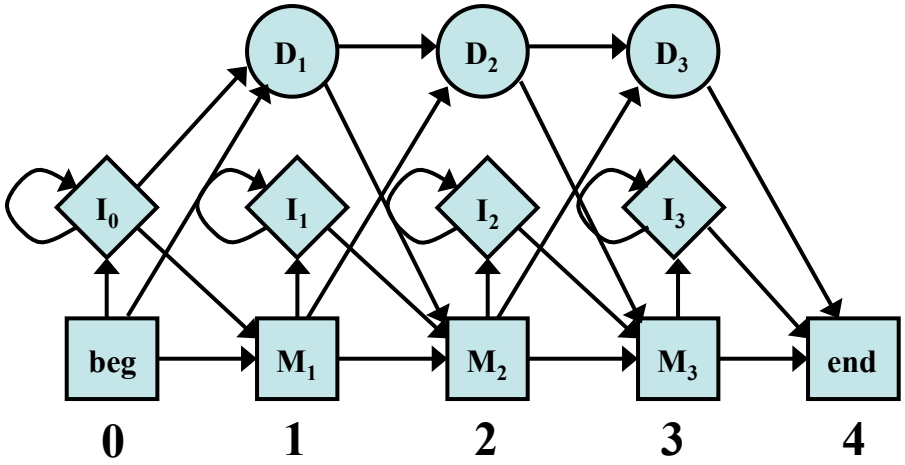
  - length of the model: heuristics or systematic way (e.g., using the MAP algorithm)

Deriving HMMs from multiple alignments

(a) Multiple alignment:

```
                x x . . . x
bat             A G - - - C
rat             A G A G - C
cat             A G - A A G
gnat            - G A A A C
goat            A G - - - C
Matching        1 2 . . . 3
```

(b) Profile-HMM architecture:

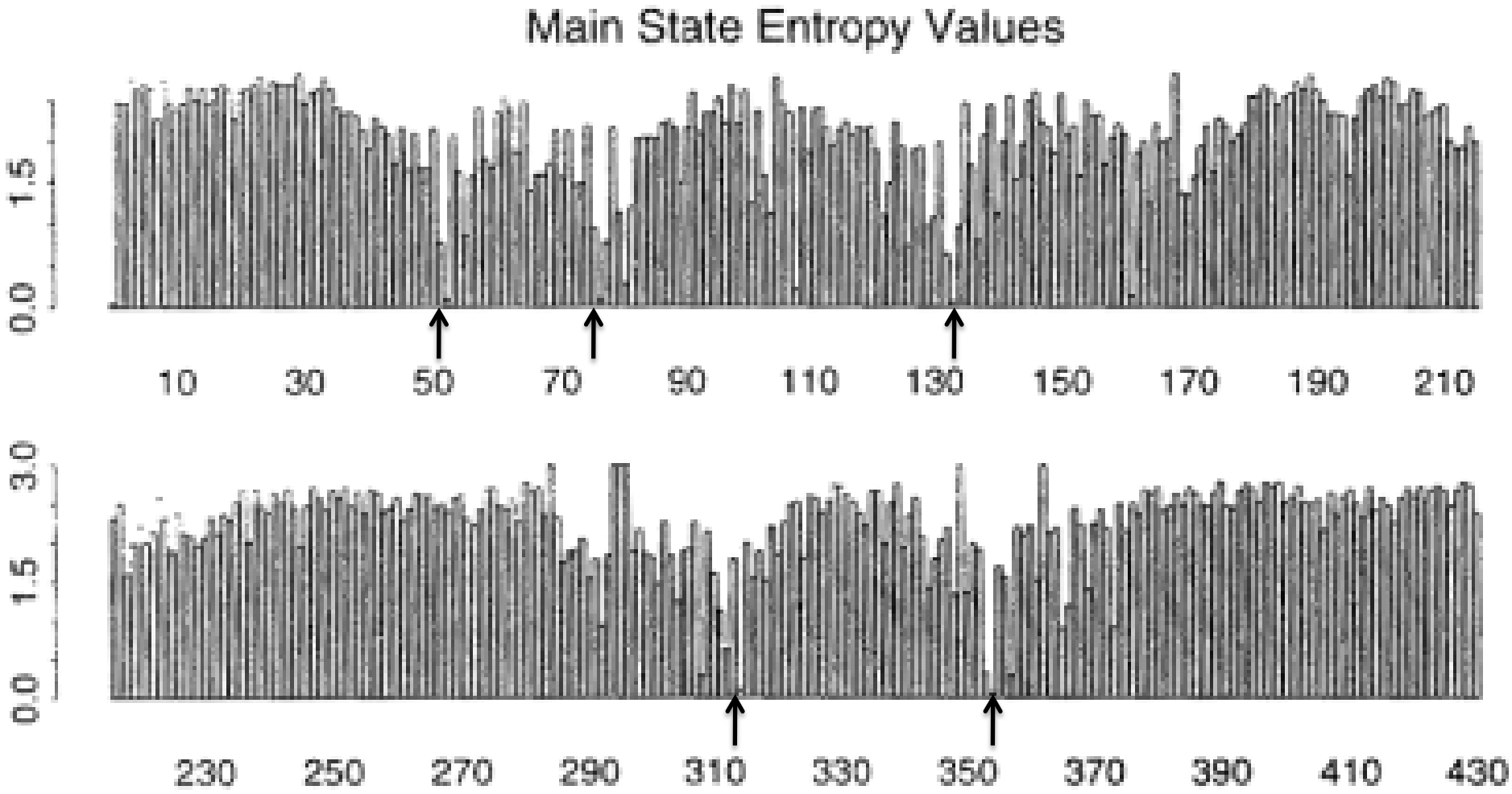


(c) Observed emission/transition counts

		0	1	2	3
Emission from M	A	-	4	0	0
	C	-	0	0	4
	G	-	0	5	1
	T	-	0	0	0
Emission from I	A	0	0	6	0
	C	0	0	0	0
	G	0	0	1	0
	T	0	0	0	0
Transition probabilities	M-M	4	4	2	4
	M-D	1	0	0	0
	M-I	0	0	3	0
	I-M	0	0	2	0
	I-I	0	0	4	0
	D-M	-	1	0	0
	D-D	-	0	0	0

A simple rule that works well is that columns that are more than half gap characters should be modeled by inserts.

Sequence conservation: entropy of the emission probability distributions



High entropy indicates non-conserved positions (note: here negative entropy a plotted.

Matching a sequence to a profile HMM (global alignment)

- Viterbi algorithm: pHMM θ (with L matching states) and a query sequence $x_1 x_2 .. x_N$

$$V_j^{\mathrm{M}}(i) = e_{\mathrm{M}_j}(x_i) \cdot \max \begin{cases} V_{j-1}^{\mathrm{M}}(i-1) \cdot a_{\mathrm{M}_{j-1}\mathrm{M}_j}, \\ V_{j-1}^{\mathrm{I}}(i-1) \cdot a_{\mathrm{I}_{j-1}\mathrm{M}_j}, \\ V_{j-1}^{\mathrm{D}}(i-1) \cdot a_{\mathrm{D}_{j-1}\mathrm{M}_j}; \end{cases}$$

$$V_j^{\mathrm{I}}(i) = e_{\mathrm{I}_j}(x_i) \cdot \max \begin{cases} V_j^{\mathrm{M}}(i-1) \cdot a_{\mathrm{M}_j\mathrm{I}_j}, \\ V_j^{\mathrm{I}}(i-1) \cdot a_{\mathrm{I}_j\mathrm{I}_j}; \end{cases}$$

$$V_j^{\mathrm{D}}(i) = \max \begin{cases} V_{j-1}^{\mathrm{M}}(i) \cdot a_{\mathrm{M}_{j-1}\mathrm{D}_j}, \\ V_{j-1}^{\mathrm{D}}(i) \cdot a_{\mathrm{D}_{j-1}\mathrm{D}_j}; \end{cases}$$

for i=1,…,L, j=1,…,N

Initialization:

$$V_0^M(0) = 1; \quad V_{j>0}^M(0) = 0; \quad V_0^M(i>0) = 0;$$
$$V_j^I(0) = 0;$$
$$V_0^D(i) = 0;.$$

Termination:

$$V = \max\left[V_L^M(N), V_L^I(N), V_L^D(N)\right]$$

Note: this is slightly different from the textbook; no transition from D_j to I_j or from I_j to D_{j+1}.

Viterbi algorithm: trace back

$$ptr_j^{\mathrm{M}}(i) = \begin{cases} \nwarrow \; if \quad V_j^{\mathrm{M}}(i) = e_{\mathrm{M}_j}(x_i) \cdot V_{j-1}^{\mathrm{M}}(i-1) \cdot a_{\mathrm{M}_{j-1}\mathrm{M}_j} \\ \uparrow \; if \quad V_j^{\mathrm{M}}(i) = e_{\mathrm{M}_j}(x_i) \cdot V_{j-1}^{\mathrm{I}}(i-1) \cdot a_{\mathrm{I}_{j-1}\mathrm{M}_j} \\ \leftarrow if \quad V_j^{\mathrm{M}}(i) = e_{\mathrm{M}_j}(x_i) \cdot V_{j-1}^{\mathrm{D}}(i-1) \cdot a_{\mathrm{D}_{j-1}\mathrm{M}_j} \end{cases}$$

$$ptr_j^{\mathrm{I}}(i) = \begin{cases} \nwarrow \; if \quad V_j^{\mathrm{I}}(i) = e_{\mathrm{I}_j}(x_i) \cdot V_j^{\mathrm{M}}(i-1) \cdot a_{\mathrm{M}_j\mathrm{I}_j}, \\ \uparrow \; if \quad V_j^{\mathrm{I}}(i) = e_{\mathrm{I}_j}(x_i) \cdot V_j^{\mathrm{I}}(i-1) \cdot a_{\mathrm{I}_j\mathrm{I}_j}; \end{cases}$$

$$ptr_j^{\mathrm{D}}(i) = \begin{cases} \nwarrow \; if \quad V_j^{\mathrm{D}}(i) = e_{\mathrm{D}_j}(x_i) \cdot V_{j-1}^{\mathrm{M}}(i) \cdot a_{\mathrm{M}_{j-1}\mathrm{D}_j}, \\ \rightarrow if \quad V_j^{\mathrm{D}}(i) = e_{\mathrm{D}_j}(x_i) \cdot V_{j-1}^{\mathrm{D}}(i) \cdot a_{\mathrm{D}_{j-1}\mathrm{D}_j}; \end{cases}$$

$$ptr_{ter} = \begin{cases} M \quad if \quad V = V_L^M(N) \\ I \quad if \quad V = V_L^I(N) \\ D \quad if \quad V = V_L^D(N) \end{cases}$$

PrintStateSeq(ptr^{M}, ptr^{I}, ptr^{D}, ptr$_{ter}$, N, L)

```
PrintStateSeq(ptrM, ptrI, ptrD, ptrter, i, j)
    if i=0 or j=0
        return;
    if ptrter = "M"
        if ptrM(L) = "↖"
            printStateSeq(ptrM, ptrI, ptrD, "M", i-1, j-1)
        else if ptrM(L) = "↑"
            printStateSeq(ptrM, ptrI, ptrD, "I", i-1, j-1)
        else if ptrM(L) = "←"
            printStateSeq(ptrM, ptrI, ptrD, "D", i-1, j-1)
        print Mj
    else if ptrter = "I"
        if ptrM(L) = "↖"
            printStateSeq(ptrM, ptrI, ptrD, "M", i-1, j)
        else if ptrM(L) = "↑"
            printStateSeq(ptrM, ptrI, ptrD, "I", i-1, j)
        print Ij
    else
        if ptrM(L) = "↖"
            printStateSeq(ptrM, ptrI, ptrD, "M", i, j-1)
        else if ptrM(L) = "←"
            printStateSeq(ptrM, ptrI, ptrD, "D", i, j-1)
        print Dj
```

Matching a sequence to a profile HMM (global alignment)

- Forward: pHMM θ (with L matching states) and a query sequence $x_1 x_2 .. x_N$

$$F_j^{\mathrm{M}}(i) = e_{\mathrm{M}_j}(x_i) \cdot \left[F_{j-1}^{\mathrm{M}}(i-1) \cdot a_{\mathrm{M}_{j-1}\mathrm{M}_j} + F_{j-1}^{\mathrm{I}}(i-1) \cdot a_{\mathrm{I}_{j-1}\mathrm{M}_j} + F_{j-1}^{\mathrm{D}}(i-1) \cdot a_{\mathrm{D}_{j-1}\mathrm{M}_j} \right]$$

$$F_j^{\mathrm{I}}(i) = e_{\mathrm{I}_j}(x_i) \cdot \left[F_j^{\mathrm{M}}(i-1) \cdot a_{\mathrm{M}_j\mathrm{I}_j} + F_j^{\mathrm{I}}(i-1) \cdot a_{\mathrm{I}_j\mathrm{I}_j} \right]$$

$$F_j^{\mathrm{D}}(i) = F_{j-1}^{\mathrm{M}}(i) \cdot a_{\mathrm{M}_{j-1}\mathrm{D}_j} + F_{j-1}^{\mathrm{D}}(i) \cdot a_{\mathrm{D}_{j-1}\mathrm{D}_j}$$

for i=1,…,L, j=1,…,N

Initialization:

$$F_0^M(0) = 1; \quad F_{j>0}^M(0) = 0; \quad F_0^M(i>0) = 0;$$

$$F_j^I(0) = 0;$$

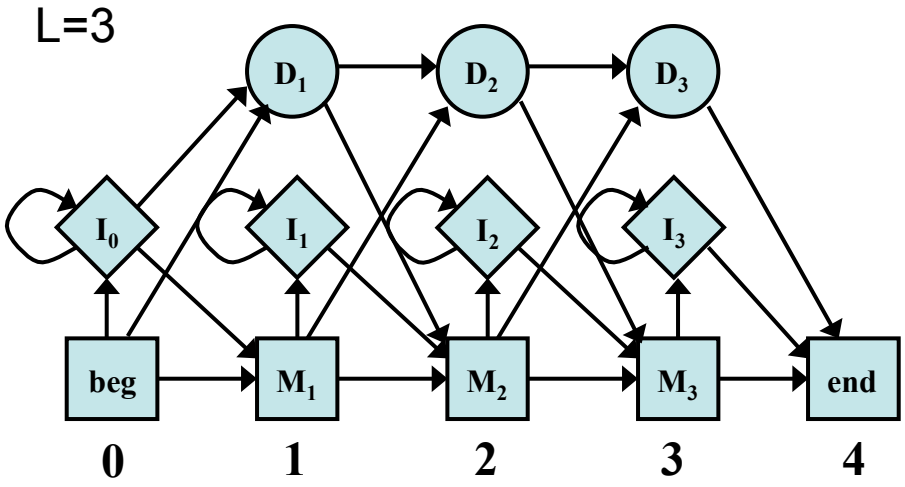
$$F_0^D(i) = 0;$$

Termination:

$$F = F_L^M(N) + F_L^I(N) + F_L^D(N)$$

Note: this is slightly different from the textbook; no transition from D_j to I_j or from I_j to D_{j+1}.

Example



Query: AGG (N=3)

		0	1	2	3
Emission from M	A	-	1	.5	0
	C	-	0	0	.8
	G	-	0	.5	.2
	T	-	0	0	0
Emission from I	A		.2	.2	.9	.2
	C		.3	.3	0	.3
	G		.3	.3	.1	.3
	T		.2	.2	0	.2
Transition probabilities	M-M		.8	1	.4	1
	M-D		.2	0	0	0
	M-I		0	0	.6	0
	I-M		.5	.5	.3	.5
	I-I		.5	.5	.7	.5
	D-M		-	1	.5	1
	D-D		-	0	.5	0

Example: Viterbi algorithm

Emission from M

Emission from I

Transition probabilities

	0	1	2	3
A	-	1	.5	0
C	-	0	0	.8
G	-	0	.5	.2
T	-	0	0	0
A	.2	.2	.9	.2
C	.3	.3	0	.3
G	.3	.3	.1	.3
T	.2	.2	0	.2
M-M	.8	1	.4	1
M-D	.2	0	0	0
M-I	0	0	.6	0
I-M	.5	.5	.3	.5
I-I	.5	.5	.7	.5
D-M	-	1	.5	1
D-D	-	0	.5	0

Initialization

j	i	0	1	2	3
0	V^M	1	0	0	0
	V^I	0			
	V^D	0	0	0	0
1	V^M	0			
	V^I	0			
	V^D				
2	V^M	0			
	V^I	0			
	V^D				
3	V^M	0			
	V^I	0			
	V^D				

Query: X=AGG (N=3)

Example: Viterbi algorithm

Emission from M

Emission from I

Transition probabilities

	0	1	2	3
A	-	1	.5	0
C	-	0	0	.8
G	-	0	.5	.2
T	-	0	0	0
A	.2	.2	.9	.2
C	.3	.3	0	.3
G	.3	.3	.1	.3
T	.2	.2	0	.2
M-M	.8	1	.4	1
M-D	.2	0	0	0
M-I	0	0	.6	0
I-M	.5	.5	.3	.5
I-I	.5	.5	.7	.5
D-M	-	1	.5	1
D-D	-	0	.5	0

j

0

1

2

3

i	0	1	2	3
V^M	1	0	0	0
V^I	0			
V^D	0	0	0	0
V^M	0			
V^I	0			
V^D	.2			
V^M	0			
V^I	0			
V^D				
V^M	0			
V^I	0			
V^D				

Query: X=AGG (N=3)

$$V_1^D(0) = \max\left(V_0^M(0)a_{M_0D_1}, V_0^D(0)a_{D_0D_1}\right)$$
$$= \max(0.2,0) = 0.2$$

Example: Viterbi algorithm

Emission from M

Emission from I

Transition probabilities

	0	1	2	3
A	-	1	.5	0
C	-	0	0	.8
G	-	0	.5	.2
T	-	0	0	0
A	.2	.2	.9	.2
C	.3	.3	0	.3
G	.3	.3	.1	.3
T	.2	.2	0	.2
M-M	.8	1	.4	1
M-D	.2	0	0	0
M-I	0	0	.6	0
I-M	.5	.5	.3	.5
I-I	.5	.5	.7	.5
D-M	-	1	.5	1
D-D	-	0	.5	0

j

i	0	1	2	3
V^M	1	0	0	0
V^I	0	0		
V^D	0	0	0	0
V^M	0	.8		
V^I	0			
V^D	.2			
V^M	0			
V^I	0			
V^D	0			
V^M	0			
V^I	0			
V^D	0			

Query: X=AGG (N=3)

$$V_1^M(1) = e_{M_1}(x_1) \cdot \max\left(V_0^M(0)a_{M_0M_1}, V_0^I(0)a_{I_0M_1}, V_0^D(0)a_{D_0M_1}\right)$$
$$= 1 \times \max(1 \times 0.8, 0 \times 0.5, 0 \times 1) = 0.8$$

Example: Viterbi algorithm

	0	1	2	3
A	-	1	.5	0
C	-	0	0	.8
G	-	0	.5	.2
T	-	0	0	0
A	.2	.2	.9	.2
C	.3	.3	0	.3
G	.3	.3	.1	.3
T	.2	.2	0	.2
M-M	.8	1	.4	1
M-D	.2	0	0	0
M-I	0	0	.6	0
I-M	.5	.5	.3	.5
I-I	.5	.5	.7	.5
D-M	-	1	.5	1
D-D	-	0	.5	0

Emission from M

Emission from I

Transition probabilities

j	i	0	1	2	3
0	V^M	1	0	0	0
	V^I	0	0	0	0
	V^D	0	0	0	0
1	V^M	0	.8	0	0
	V^I	0	0	0	0
	V^D	.2	0	0	0
2	V^M	0	.1	.4	0
	V^I	0	0	.006	.0240
	V^D	0	0	0	0
3	V^M	0	0	.008	**.0320**
	V^I	0	0	0	.0014
	V^D	0	0	0	0

Query: X=AGG (N=3)

$$V_3^M(3) = ?$$

Example: Viterbi algorithm

Emission from M

Emission from I

Transition probabilities

	0	1	2	3
A	-	1	.5	0
C	-	0	0	.8
G	-	0	.5	.2
T	-	0	0	0
A	.2	.2	.9	.2
C	.3	.3	0	.3
G	.3	.3	.1	.3
T	.2	.2	0	.2
M-M	.8	1	.4	1
M-D	.2	0	0	0
M-I	0	0	.6	0
I-M	.5	.5	.3	.5
I-I	.5	.5	.7	.5
D-M	-	1	.5	1
D-D	-	0	.5	0

j	i	0	1	2	3
0	V^M	1	0	0	0
	V^I	0	0	0	0
	V^D	0	0	0	0
1	V^M	0	.8	0	0
	V^I	0	0	0	0
	V^D	.2	0	0	0
2	V^M	0	.1	.4	0
	V^I	0	0	.006	.0240
	V^D	0	0	0	0
3	V^M	0	0	.008	**.0320**
	V^I	0	0	0	.0014
	V^D	0	0	0	0

Traceback

Query: X=AGG (N=3)

Searching with profile HMMs

- Main usage of profile HMMs
  - Detecting potential sequences in a family
  - Core algorithm: matching a sequence to a profile HMMs
    - Viterbi algorithm or forward algorithm
  - Comparing the resulting probability with random model (R): log-odd score

$$P(x \mid R) = \prod_i q_{x_i}$$

where q_x is the frequency of observing x_i.

Matching a sequence to a profile HMM (global alignment)

Viterbi algorithm

$$V_j^{\mathrm{M}}(i) = \log \frac{e_{\mathrm{M}_j}(x_i)}{q_{x_i}} + \max\begin{cases} \log V_{j-1}^{\mathrm{M}}(i-1) + \log a_{\mathrm{M}_{j-1}\mathrm{M}_j}, \\ \log V_{j-1}^{\mathrm{I}}(i-1) + \log a_{\mathrm{I}_{j-1}\mathrm{M}_j}, \\ \log V_{j-1}^{\mathrm{D}}(i-1) + \log a_{\mathrm{D}_{j-1}\mathrm{M}_j}; \end{cases}$$

$$V_j^{\mathrm{I}}(i) = \log \frac{e_{\mathrm{I}_j}(x_i)}{q_{x_i}} + \max\begin{cases} \log V_j^{\mathrm{M}}(i-1) + \log a_{\mathrm{M}_j\mathrm{I}_j}, \\ \log V_j^{\mathrm{I}}(i-1) + \log a_{\mathrm{I}_j\mathrm{I}_j}; \end{cases}$$

$$V_j^{\mathrm{D}}(i) = \max\begin{cases} \log V_{j-1}^{\mathrm{M}}(i) + \log a_{\mathrm{M}_{j-1}\mathrm{D}_j}, \\ \log V_{j-1}^{\mathrm{D}}(i) + \log a_{\mathrm{D}_{j-1}\mathrm{D}_j}; \end{cases}$$

for i=1,…,L, j=1,…,N

Initialization:

$V_0^M(0) = 0; \quad V_{j>0}^M(0) = -\infty; \quad V_0^M(i>0) = -\infty;$

$V_j^I(0) = -\infty;$

$V_0^D(i) = -\infty;.$

Termination:

$$V = \max\left[V_L^M(N), V_L^I(N), V_L^D(N)\right]$$

Matching a sequence to a profile HMM (global alignment)

Forward algorithm

$$F_j^{\mathrm{M}}(i) = \log\frac{e_{\mathrm{M}_j}(x_i)}{q_{x_i}} + \log[a_{\mathrm{M}_{j-1}\mathrm{M}_j}\exp(F_{j-1}^{\mathrm{M}}(i-1)) + a_{\mathrm{I}_{j-1}\mathrm{M}_j}\exp(F_{j-1}^{\mathrm{I}}(i-1)) + a_{\mathrm{D}_{j-1}\mathrm{M}_j}\exp(F_{j-1}^{\mathrm{D}}(i-1))];$$

$$F_j^{\mathrm{I}}(i) = \log\frac{e_{\mathrm{I}_j}(x_i)}{q_{x_i}} + \log[a_{\mathrm{M}_j\mathrm{I}_j}\exp(F_j^{\mathrm{M}}(i-1)) + a_{\mathrm{I}_j\mathrm{I}_j}\exp(F_j^{\mathrm{I}}(i-1))];$$

$$F_j^{\mathrm{D}}(i) = \log[a_{\mathrm{M}_{j-1}\mathrm{D}_j}\exp(F_{j-1}^{\mathrm{M}}(i)) + a_{\mathrm{I}_{j-1}\mathrm{D}_j}\exp(F_{j-1}^{\mathrm{I}}(i))];$$

Initialization:

$$V_0^M(0) = 0; \quad V_{j>0}^M(0) = -\infty; \quad V_0^M(i>0) = -\infty;$$

$$V_0^I(0) = -\infty;$$

$$V_0^D(i) = -\infty.$$

Termination:

$$F = \log\left[\exp\left(F_L^M(N)\right) + \exp\left(F_L^I(N)\right) + \exp\left(F_L^D(N)\right)\right]$$

Significance of HMM alignment

- The log-odd score of local Viterbi alignment (V) alignment between a random sequence and a profile HMM follows a Gumbel (type I EVD) distribution

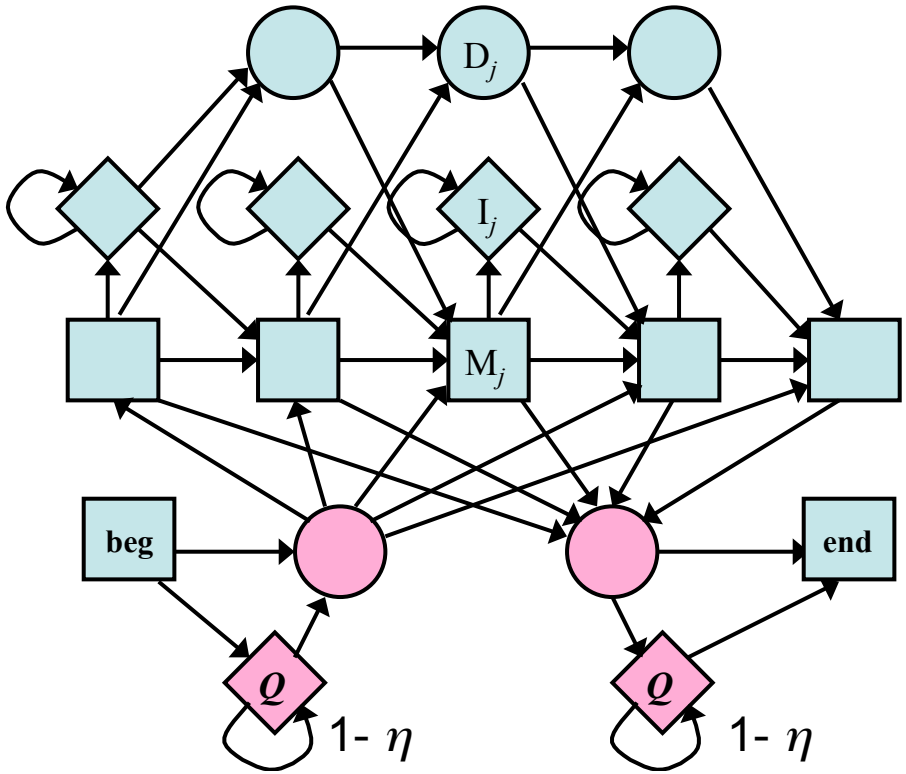
$$P(V \geq t) = 1 - \exp\left[-e^{-\lambda(t-\mu)}\right]$$

- With ~200 Viterbi, the location parameter μ can be accurately estimated;
- λ~log(z), z is the base of the log-odd score, e.g., z=2 when the sequence length approaches infinite
- The length effect can be corrected by $\lambda \sim \log 2 + \frac{1.44}{hN}$
  - Where, N is the length, and h is the average relative entropy per match state in the pHMM;
  - For typical Pfam models, N~140, h~1.8, λ~log2+0.0057, a small correction.

Eddy, PLoS Comp. Biol., 4:1, 2008

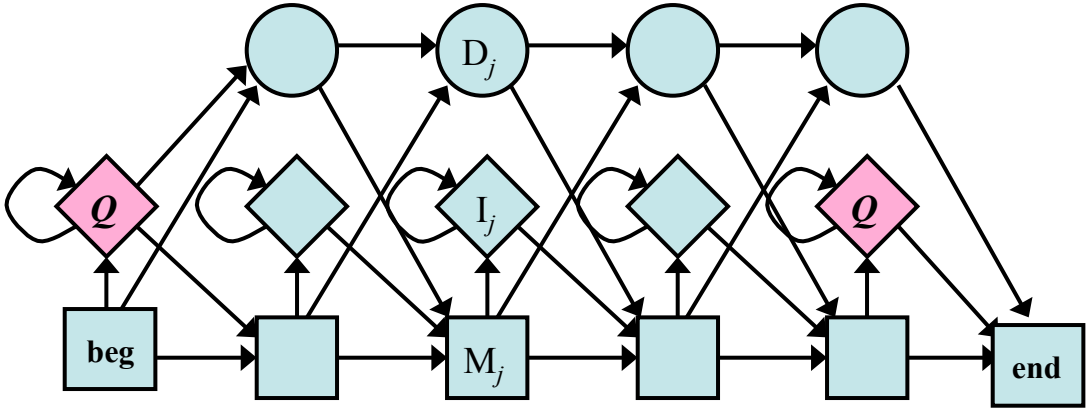
Variants for non-global alignments

- Local alignment (Smith-Waterman type)
  - Emission prob. in flanking states use background values q_x.
  - Looping prob. close to 1, e.g. (1- η) for some small η.



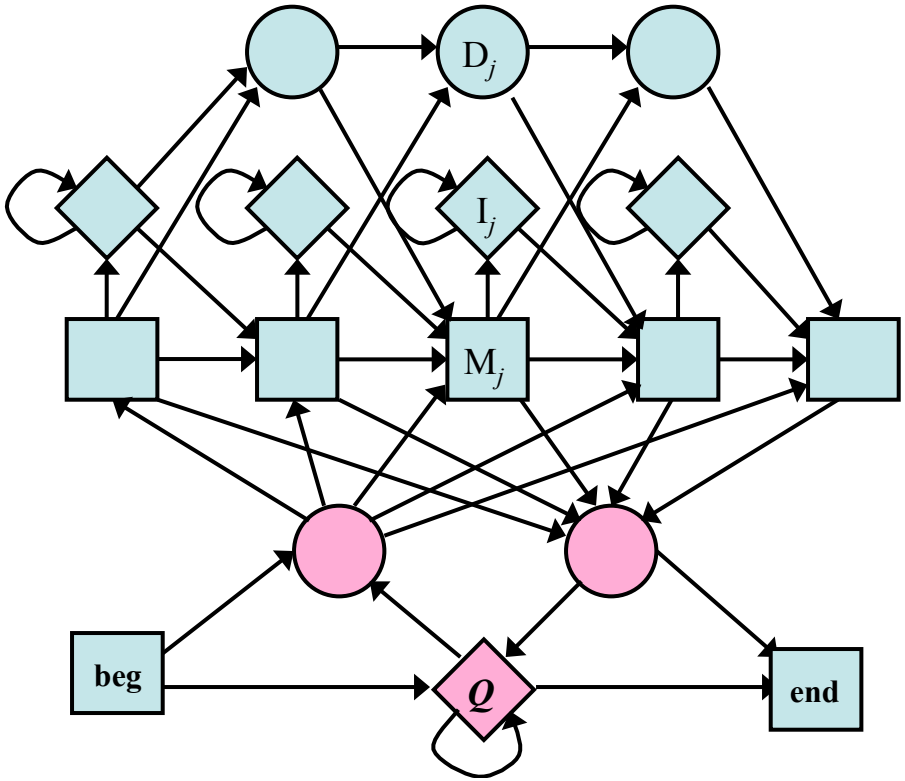
Variants for non-global alignments

- Overlap (also called *glocal* or *fit*) alignment
  - The loop probability of the *first* and *last* insert states is much higher than the other insert states
  - When expecting to find either present as a whole or absent (e.g., of a protein domain within a protein)
  - Transition to first delete state allows missing first residue



Variants for non-global alignments

- Repeat alignments
  - Transition from right flanking state back to random model
  - Can find multiple matching segments in query string

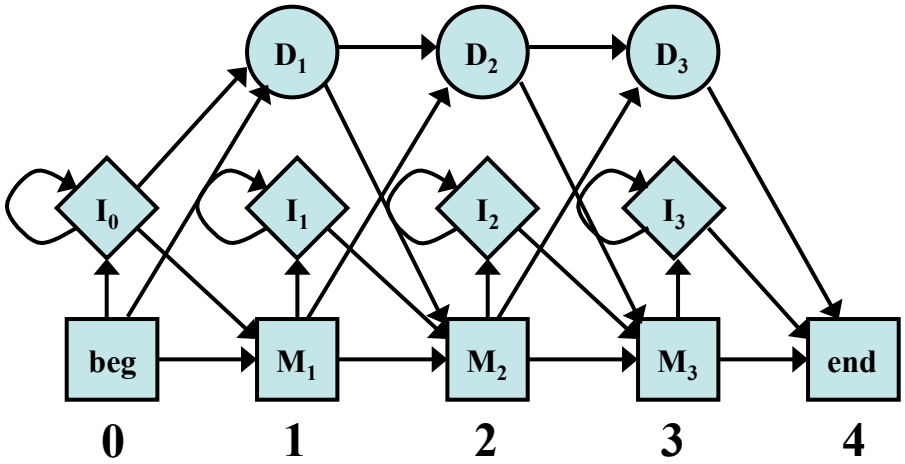


Optimal model construction: different ways of marking columns

(a) Multiple alignment:

```
                x x . . . x
bat             A G - - - C
rat             A - A G - C
cat             A G - A A -
gnat            - - A A A C
goat            A G - - - C
Matching        1 2 . . . 3
```

(b) Profile-HMM architecture:



(c) Observed emission/transition counts

		0	1	2	3
Emission from M	A	-	4	0	0
	C	-	0	0	4
	G	-	0	3	0
	T	-	0	0	0
Emission from I	A	0	0	6	0
	C	0	0	0	0
	G	0	0	1	0
	T	0	0	0	0
Transition probabilities	M-M	4	3	2	4
	M-D	1	1	0	0
	M-I	0	0	1	0
	I-M	0	0	2	0
	I-D	0	0	1	0
	I-I	0	0	4	0
	D-M	-	0	0	1
	D-D	-	1	0	0
	D-I	-	0	2	0

Optimal model construction

- MAP (match-insert assignment)
  - Recursive calculation of a number S_j
    - S_j: log prob. of the optimal model for alignment up to and including column j, assuming j is marked.
    - S_j is calculated from S_i and summed log prob. between i and j.
    - T_{ij}: summed log prob. of all the state transitions between marked i and j.

$$T_{ij} = \sum_{x,y \in \{\mathrm{M,D,I}\}} c_{xy} \log a_{xy}$$

      - c_{xy} are obtained from partial state paths implied by marking i and j.

Optimal model construction

- Algorithm: MAP model construction
  - Initialization:
    - $S_0 = 0,\ M_{L+1} = 0.$
  - Recurrence: for $j = 1,..., L+1$:

$$S_j = \max_{0 \le i < j} \left(S_i + T_{ij} + M_j + I_{i+1,j-1} + \lambda \right)$$

$$\sigma_j = \arg\max_{0 \le i < j} \left(S_i + T_{ij} + M_j + I_{i+1,j-1} + \lambda \right)$$

  - Traceback: from $j = \sigma_{L+1}$, while $\sigma_j > 0$:
    - Mark column j as a match column
    - j ← σ_j.

Weighting training sequences

- Input sequences are random?
- “Assumption: all examples are independent samples” might be incorrect
- Solutions
  - Weight sequences based on similarity: highly similar pair of training sequences receive lower weights

Multiple sequence alignment (MSA) by training profile HMM

- Sequence profiles can be represented as probabilistic models like profile HMMs.
  - ML methods for building (training) profile HMM are based on multiple sequence alignment
  - Profile HMMs can also be trained from initially unaligned sequences using the Baum-Welch-like EM algorithm
    - Simultaneously aligning multiple sequences and building the profile HMM from the multiple alignment

Multiple alignment with a known profile HMM

- A step backward: to derive a multiple alignment from a known profile HMM model
  - e.g., to align many sequences from the same family based on the HMM model built from the (seed) multiple alignment of a small representative set of sequences in the family.
- It just requires calculating a Viterbi alignment for each individual sequence
  - Match a sequence to a profile HMM: Viterbi algorithm
  - Residues aligned to the same match state in the profile HMM should be aligned in the same columns;
  - Given a preliminary alignment, HMM can align additional sequences.

Multiple alignment with a known profile HMM

- Comparing with other MSA program
  - Profile HMM does not align inserts whereas other MSA algorithms align the whole sequences.

Position	1	2	3	4	5	6	insert	7	8	9	10	11
	F	P	H	F	–	D	LS	H	G	S	A	Q
	F	E	S	F	G	D	LSTPDAV	M	G	N	P	K
	F	D	R	F	K	H	LKTEAEM	K	A	S	E	D
	F	T	Q	F	A	G	KDLESI	K	G	T	A	P
	F	P	K	F	K	G	LTTADQL	K	K	S	A	D
	F	S	–	F	L	K	GTSEVP	Q	N	N	P	E
	F	G	–	F	S	G	AS	–	–	D	P	G

Training profile HMM from unaligned sequences

- Simultaneously aligning multiple sequences and building the profile HMM from the multiple alignment
  - Initialization: choose the length of the profile HMM and initialize parameters of the model
  - MSA: align all sequences to the final model using the Viterbi algorithm and build a multiple alignment as described in the previous section.
  - Training: estimate the model using the Baum-Welch algorithm
  - Iterating until the model (and the MSA) converges

Profile HMM training from unaligned sequences

- Initial Model
  - The only decision that must be made in choosing an initial structure for Baum-Welch estimation is the length of the model M.
  - A commonly used rule is to set M be the average length of the training sequence.
  - We need some randomness in initial parameters to avoid local maxima.

Multiple alignment by profile HMM training

- Avoiding Local maxima
  - Baum-Welch algorithm is guaranteed to find a LOCAL maxima.
    - Models are usually quite long and there are many opportunities to get stuck in a wrong solution.
  - Solution
    - Start many times from different initial models.
    - Use some form of stochastic search algorithm, e.g. simulated annealing.

Multiple alignment by profile HMM training--Model surgery

- We can modify the model after (or during) training a model by manually checking the alignment produced from the model.
  - Some of the match states are redundant
  - Some insert states absorb too many sequences
- Model surgery
  - If a match state is used by less than ½ of training sequences, delete its module (match-insert-delete states)
  - If more than ½ of training sequences use a certain insert state, expand it into *n* new modules, where *n* is the average length of insertions
  - ad hoc, but works well

Hmmer 3

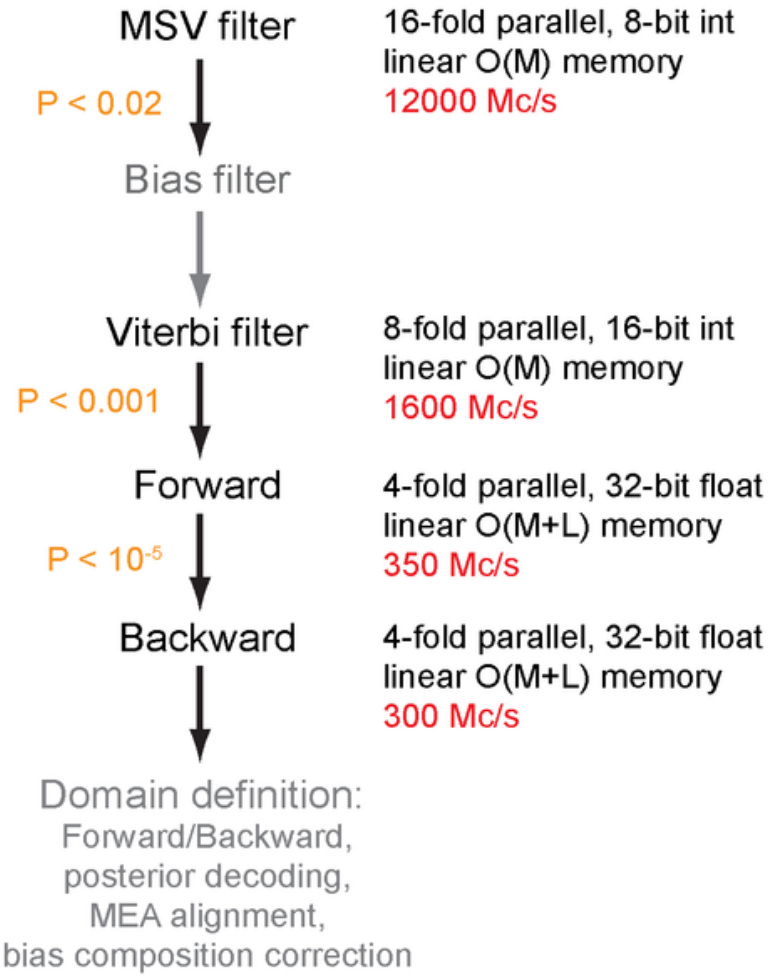


Figure 4. The HMMER3 acceleration pipeline.

Eddy SR (2011) Accelerated Profile HMM Searches. PLoS Comput Biol 7(10): e1002195. doi:10.1371/journal.pcbi.1002195
http://www.ploscompbiol.org/article/info:doi/10.1371/journal.pcbi.1002195

Accelerated Profile HMM Searches

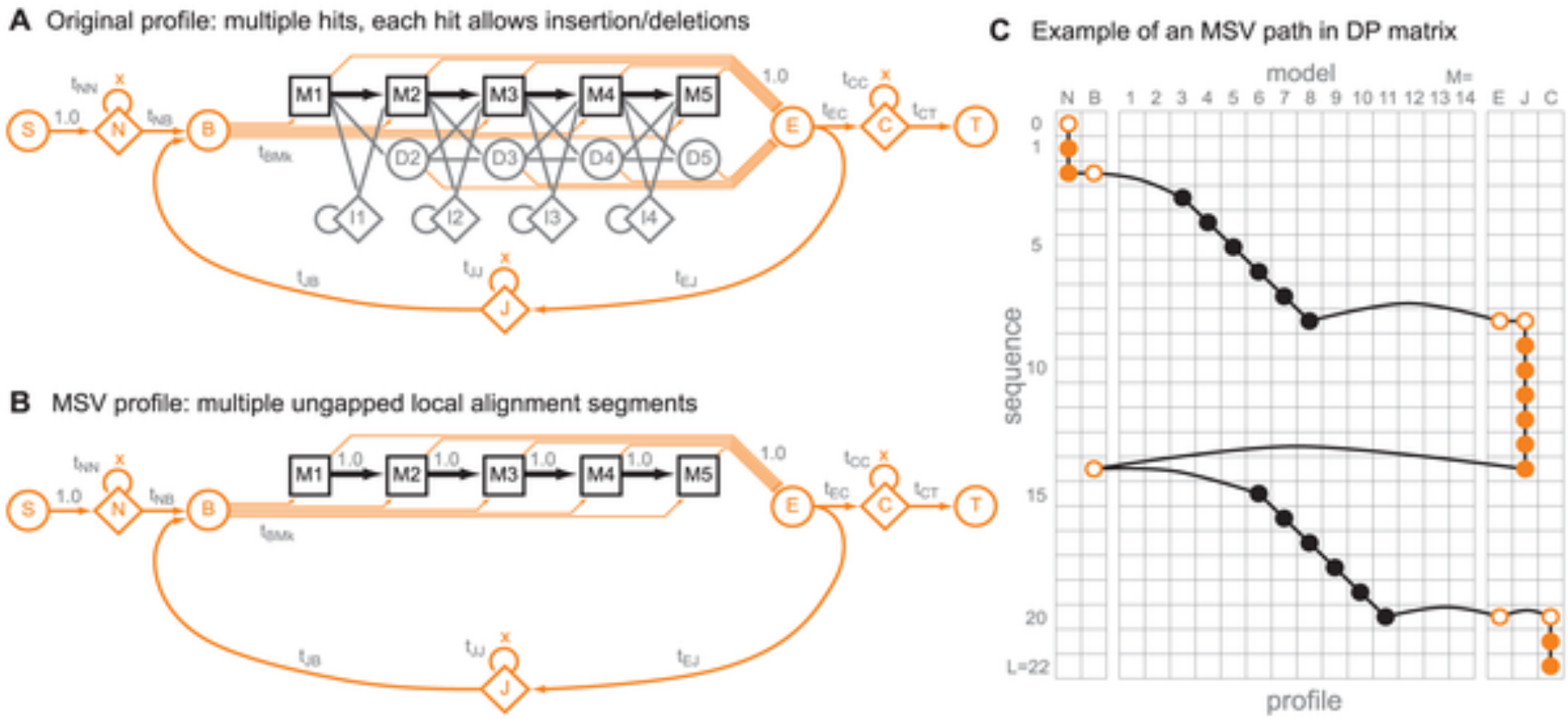


Figure 1. The MSV profile.

Eddy SR (2011) Accelerated Profile HMM Searches. PLoS Comput Biol 7(10): e1002195. doi:10.1371/journal.pcbi.1002195
http://www.ploscompbiol.org/article/info:doi/10.1371/journal.pcbi.1002195