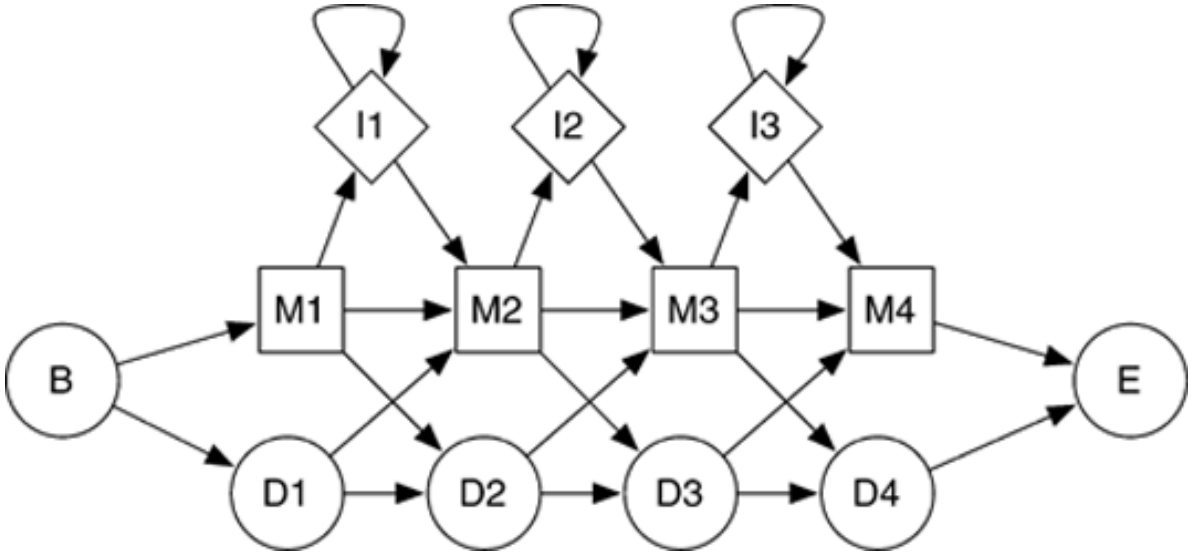
EECS730: Introduction to Bioinformatics

Lecture 07: profile Hidden Markov Model



http://bibiserv.techfak.uni-bielefeld.de/sadr2/databasesearch/hmmer/profileHMM.gif

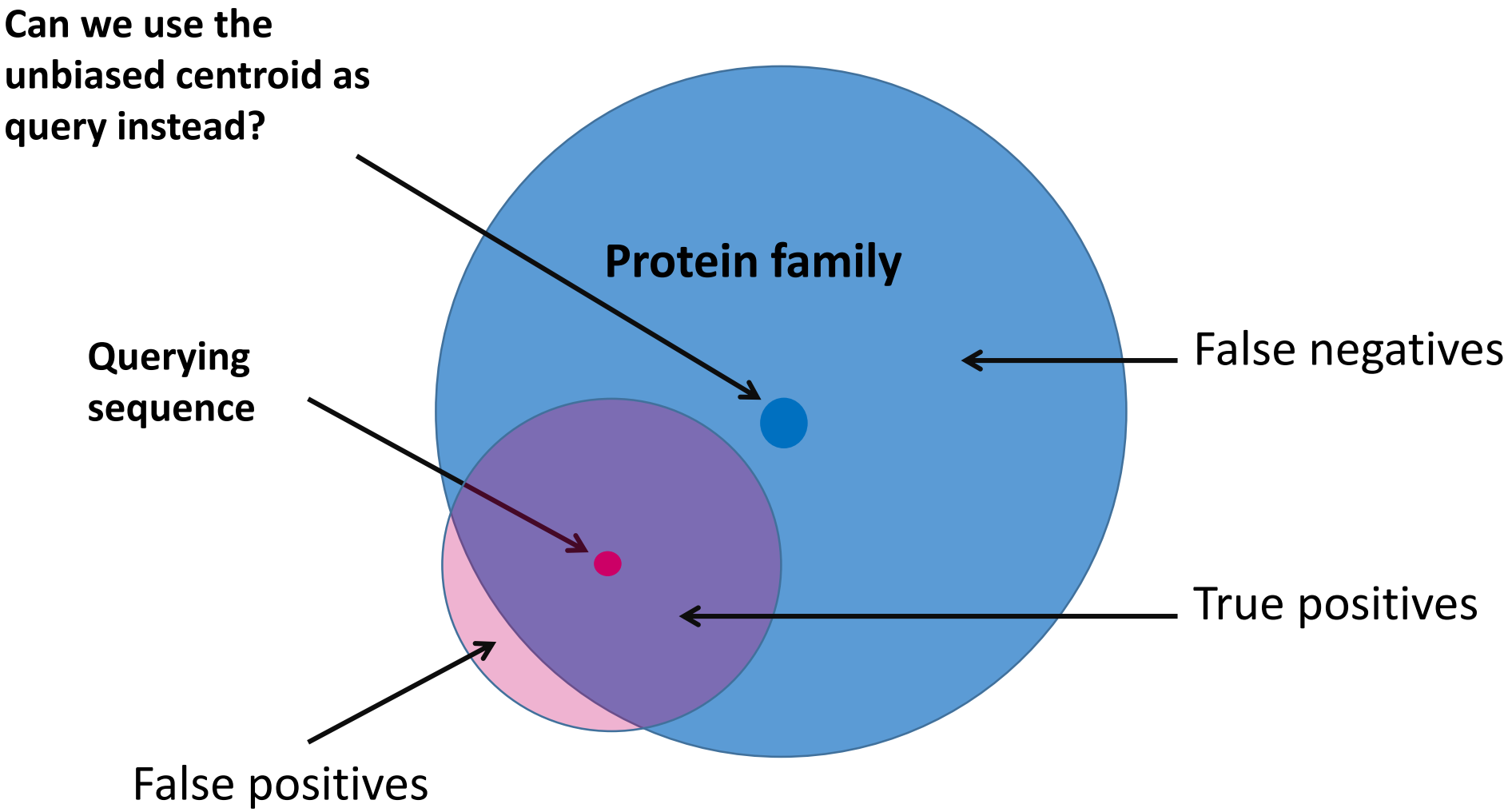
Slides adapted from Dr. Shaojie Zhang (University of Central Florida)

Information from multiple sequence alignments

- Protein/Gene family: homolog, ortholog, paralog, and xenolog

- Usually homologs are rooted form the same gene, diverged during evolution, and have similar biological functions
- Multiple alignments of homologous sequences usually reveal important sequence feature of the protein family and indicate its function
- We have discussed in the previous class how to build multiple sequence alignments from a set of homologous sequences

The revised homolog detection problem

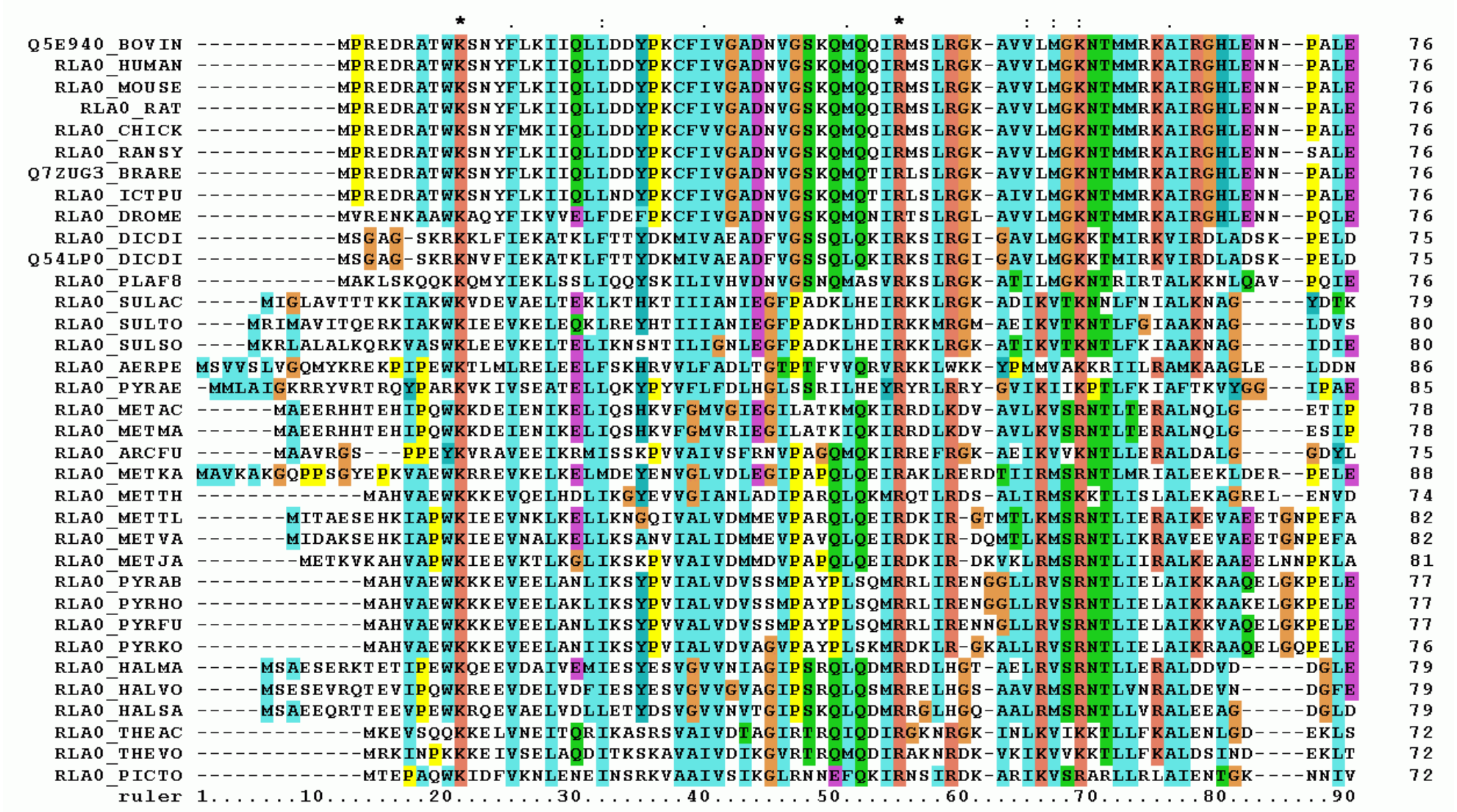


The revised homolog detection problem

- Input: a set of homologous sequences from the same protein family, and a unannotated protein sequence
- Output: the likelihood that the unannotated protein sequence is also from the protein family

- Naïve solution: perform pairwise alignment between each sequence in the family with the unannotated protein sequence
- It could be very slow, and it may not reflect true homology

Can we summarize information of a protein family from MSA?



An intuitive way is to summarize column-wise frequency

GAGGTAAAC

TCCGTAAGT

CAGGTTGGA

ACAGTCAGT

TAGGTCATT

TAGGTACTG

ATGGTAACT

CAGGTATAC

TGTGTGAGT

AAGGTAAGT

$$M_{k,j} = \frac{1}{N} \sum_{i=1}^{N} I(X_{i,j} = k),$$

$$M = \begin{matrix} A \\ C \\ G \\ T \end{matrix} \begin{bmatrix} 3 & 6 & 1 & 0 & 0 & 6 & 7 & 2 & 1 \\ 2 & 2 & 1 & 0 & 0 & 2 & 1 & 1 & 2 \\ 1 & 1 & 7 & 10 & 0 & 1 & 1 & 5 & 1 \\ 4 & 1 & 1 & 0 & 10 & 1 & 1 & 2 & 6 \end{bmatrix}$$

$$M = \begin{matrix} A \\ C \\ G \\ T \end{matrix} \begin{bmatrix} 0.3 & 0.6 & 0.1 & 0.0 & 0.0 & 0.6 & 0.7 & 0.2 & 0.1 \\ 0.2 & 0.2 & 0.1 & 0.0 & 0.0 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.1 & 0.1 & 0.7 & 1.0 & 0.0 & 0.1 & 0.1 & 0.5 & 0.1 \\ 0.4 & 0.1 & 0.1 & 0.0 & 1.0 & 0.1 & 0.1 & 0.2 & 0.6 \end{bmatrix}$$

https://en.wikipedia.org/wiki/Position_weight_matrix

Using the Position Specific Scoring Matrix

- Modified matching scores
  - Sum($p_{i,j}$ * score(j, a))

https://upload.wikimedia.org/wikipedia/commons/8/85/LexA_gram_positive_bacteria_sequence_logo.png

- Keep the original setup for the gap penalty
- RPS-BLAST
- The gaps are not handled well, we need more advanced model to account for gaps

Introducing the Markov Model

- First-order Markov Chain

$M = (Q, \pi, a)$

Q – finite set of states, say $|Q| = n$
a – n x n transition probability matrix

$$a(i,j) = \Pr[q_{t+1}=j \mid g_t=i]$$

π – n-vector, starting probability vector $\pi(i) = \Pr[q_0=i]$
For any row of a the sum of entries = 1
$\Sigma\pi(i) = 1$

Hidden Markov Model (HMM)

Hidden Markov Model is a Markov model in which one does not observe a sequence of states but results of a function prescribed on states – in our case this is emission of a symbol (amino acid or a nucleotide).

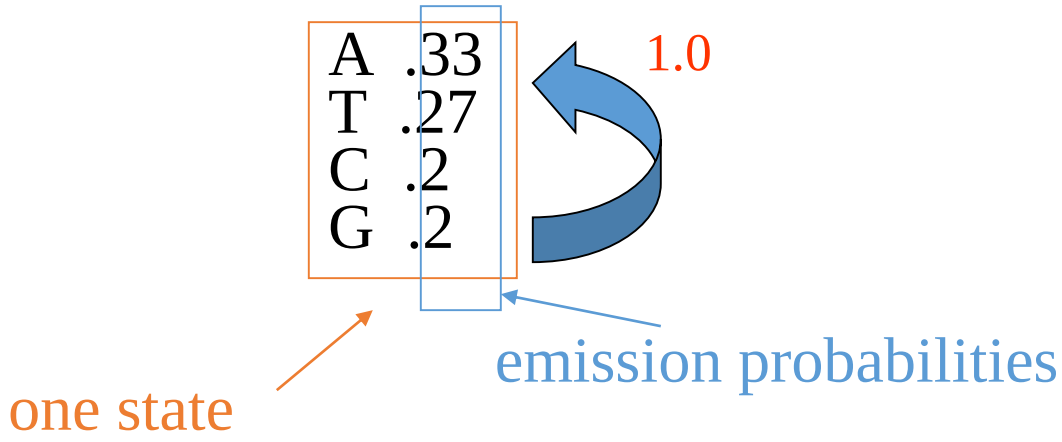
States are hidden to the observers.

Emission probabilities

- Assume that at each state a Markov process emits (with some distribution) a symbol from alphabet Σ.
- Rather than observing a sequence of states we observe a sequence of emitted symbols.

Example:

Σ ={A,C,T,G}. Generate a sequence where A,C,T,G have frequency p(A) =.33, p(G)=.2, p(C)=.2, p(T) = .27 respectively



HMM

HMM is a Markov process that at each time step generates a symbol from some alphabet, Σ, according to emission probability that depends on state.

$M = (Q, \Sigma, \pi, a, e)$

Q – finite set of states, say n states $=\{q_0, q_1, \ldots\}$

a – n x n transition probability matrix: $a(i,j) = Pr[q_{t+1}=j \mid g_t=i]$

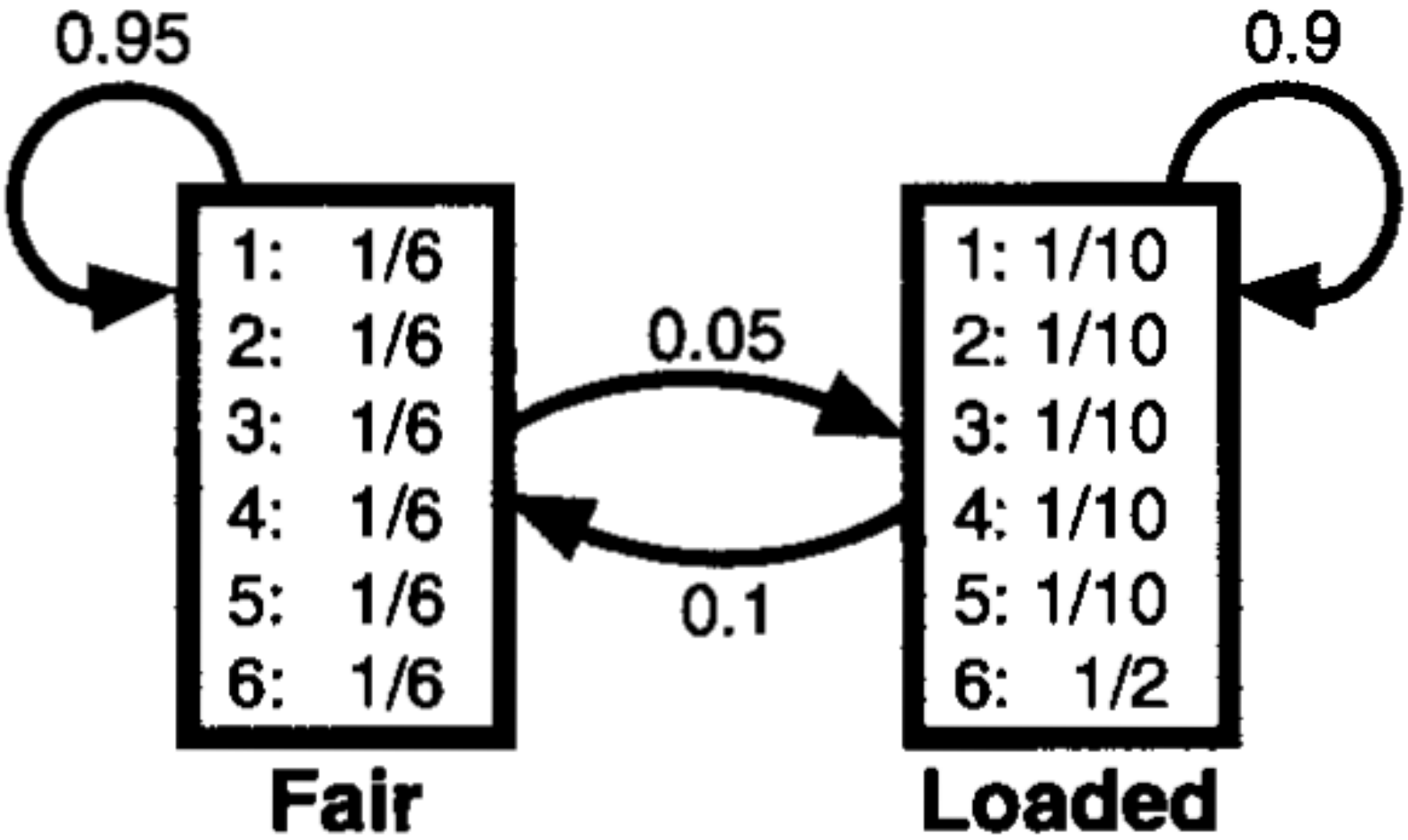
π – n-vector, start probability vector: $\pi(i) = Pr[q_0=i]$

$\Sigma = \{\sigma_1, \ldots, \sigma_k\}$-alphabet

$e(i,j) = Pr[o_t=\sigma_j \mid q_t = i]$; o_t – t^{th} element of generated sequences

= probability of generating o_j in state q_i ($S=o_0,\ldots o_T$ the output sequence)

Occasionally dishonest casino



Summarizing MSA using HMM

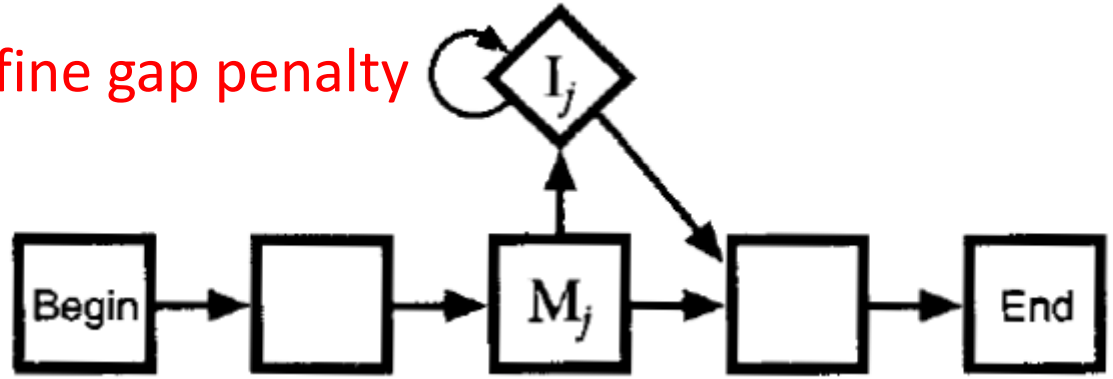
If we simply consider MSA columns without gaps
This is equivalent to PSSM

Eddy et al. *Biological Sequence Analysis*, Cambridge Press

MSA to HMM

- Considering the Insertions

Affine gap penalty

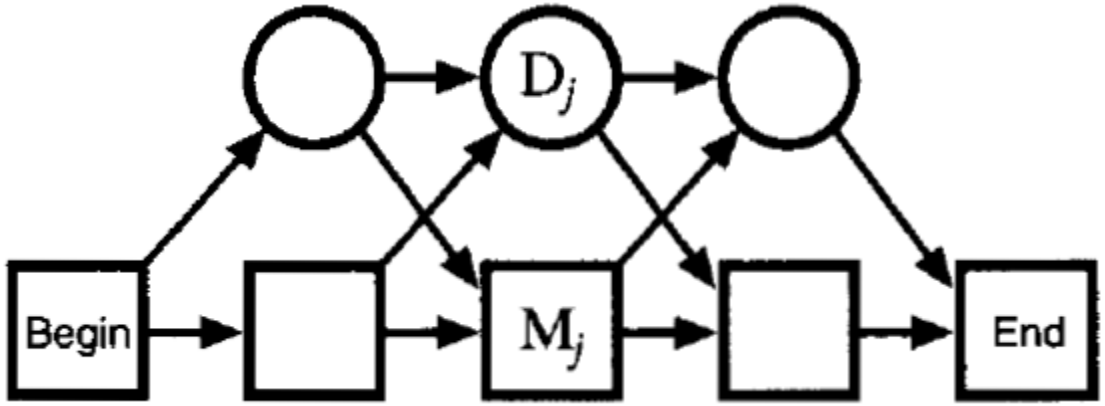


$$\log a_{M_j I_j} + \log a_{I_j M_{j+1}} + (k-1)\log a_{I_j I_j}.$$

Eddy et al. *Biological Sequence Analysis*, Cambridge Press

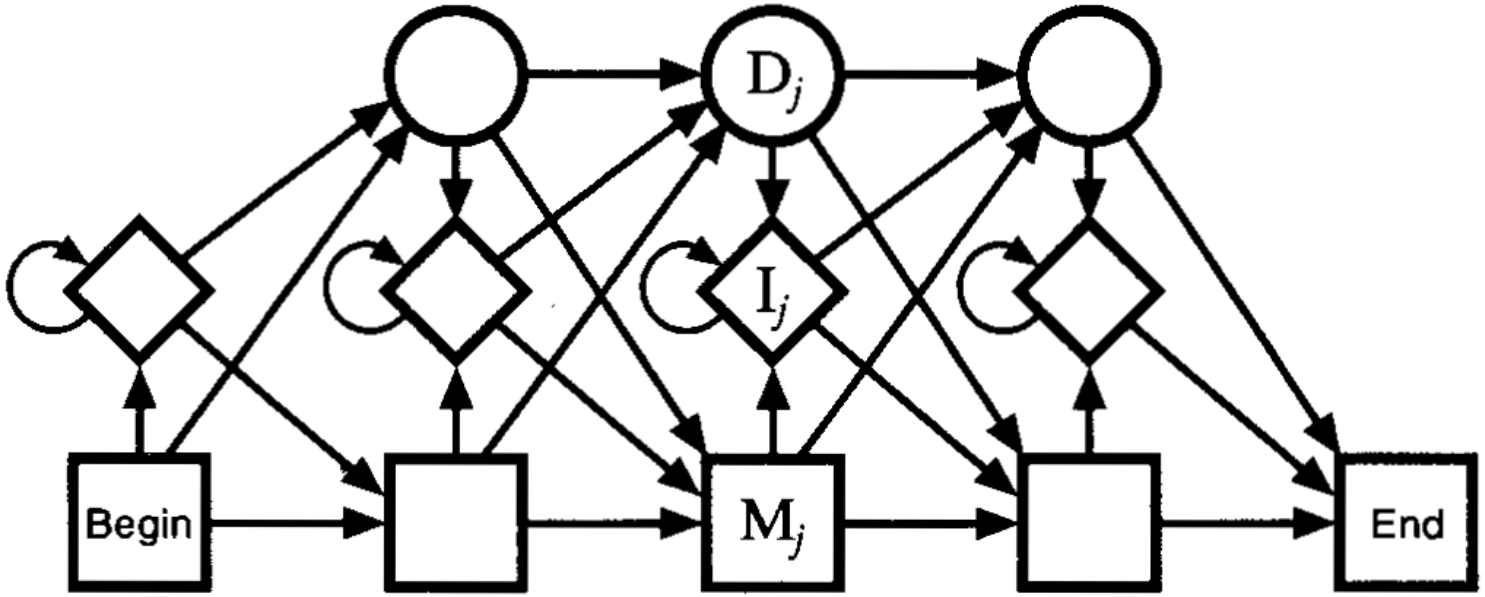
MSA to HMM

Affine gap penalty



Eddy et al. *Biological Sequence Analysis*, Cambridge Press

MSA to HMM, the complete model



Eddy et al. *Biological Sequence Analysis*, Cambridge Press

How many states should we have

- The number of matching state is usually determined as the number of columns who have non-gap majority

- Number of insertion and deletion states determined correspondingly

Computing the parameters

- Emission probability

$$e_k(a) = \frac{E_k(a)}{\sum_{a'} E_k(a')}$$

- Transition probability

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}}$$

```
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...
                  ***   *****
```

Eddy et al. *Biological Sequence Analysis*, Cambridge Press

How to align MSA profile to a sequence

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

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

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

Eddy et al. *Biological Sequence Analysis*, Cambridge Press

Time complexity

- O(MN), where M is the number of states in HMM and N is the length of the observed sequence

Viterbi algorithm for generalized HMM

Algorithm: Viterbi

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

Recursion ($i = 1 \ldots L$): $v_l(i) = e_l(x_i) \max_k(v_k(i-1)a_{kl})$;
$\text{ptr}_i(l) = \text{argmax}_k(v_k(i-1)a_{kl})$.

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

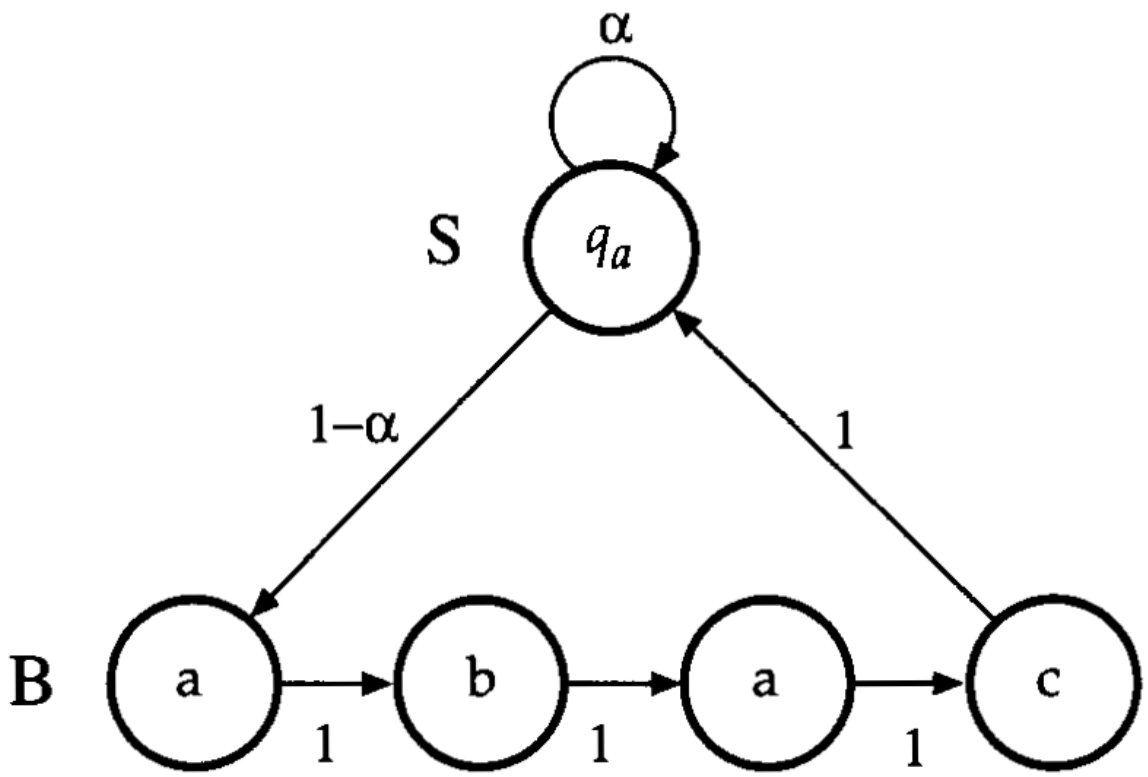
Traceback ($i = L \ldots 1$): $\pi^*_{i-1} = \text{ptr}_i(\pi^*_i)$.

Eddy et al. *Biological Sequence Analysis*, Cambridge Press

Time complexity

- $O(M^2N)$, where M is the number of states in HMM and N is the length of the observed sequence

Limitation of the Viterbi path



Eddy et al. *Biological Sequence Analysis*, Cambridge Press

Forward-backward algorithm

- Using Viterbi algorithm, we can calculate the most probable parse of the observed sequence given the HMM
- However, in many cases we want to calculate all probable parses that can give rise to the observed sequence given the HMM
- This can be very useful when there are many suboptimal paths that are nearly as good as the most probable path
- We can compute is using the Forward algorithm

Forward algorithm

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

Algorithm: Forward algorithm

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

Recursion ($i = 1 \ldots L$): $f_l(i) = e_l(x_i) \sum_k f_k(i-1) a_{kl}$.

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

Eddy et al. *Biological Sequence Analysis*, Cambridge Press

The need for decoding

- What is the probability that an observed character comes from a given state???

$$
\begin{aligned}
P(x, \pi_i = k) &= P(x_1 \ldots x_i, \pi_i = k) P(x_{i+1} \ldots x_L | x_1 \ldots x_i, \pi_i = k) \\
&= P(x_1 \ldots x_i, \pi_i = k) P(x_{i+1} \ldots x_L | \pi_i = k),
\end{aligned}
$$

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

$$
b_k(i) = P(x_{i+1} \ldots x_L | \pi_i = k).
$$

Eddy et al. *Biological Sequence Analysis*, Cambridge Press

Backward algorithm

Algorithm: Backward algorithm

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

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

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

Eddy et al. *Biological Sequence Analysis*, Cambridge Press

Decoding

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

where $P(x)$ is the result of the forward (or backward) calculation.

Eddy et al. *Biological Sequence Analysis*, Cambridge Press

PFAM

EMBL-EBI

HOME | SEARCH | BROWSE | FTP | HELP | ABOUT

Pfam 30.0 (June 2016, 16306 entries)

The Pfam database is a large collection of protein families, each represented by **multiple sequence alignments** and **hidden Markov models (HMMs)**. More...

QUICK LINKS	YOU CAN FIND DATA IN PFAM IN VARIOUS WAYS...
SEQUENCE SEARCH	Analyze your protein sequence for Pfam matches
VIEW A PFAM ENTRY	View Pfam annotation and alignments
VIEW A CLAN	See groups of related entries
VIEW A SEQUENCE	Look at the domain organisation of a protein sequence
VIEW A STRUCTURE	Find the domains on a PDB structure
KEYWORD SEARCH	Query Pfam by keywords
JUMP TO	enter any accession or ID Go Example

Enter any type of accession or ID to jump to the page for a Pfam entry or clan, UniProt sequence, PDB structure, etc.

Or view the help pages for more information

HMMER3 HMM

```
HMMER3/f [3.1 | February 2013]
NAME  globins4
LENG  149
ALPH  amino
RF    no
MM    no
CONS  yes
CS    no
MAP   yes
DATE  Thu Feb 14 16:44:36 2013
NSEQ  4
EFFN  0.964844
CKSUM 2027839109
STATS LOCAL MSV       -9.9014  0.70957
STATS LOCAL VITERBI  -10.7224  0.70957
STATS LOCAL FORWARD   -4.1637  0.70957
HMM          A        C        D        E        F        G        H        I        K        L
            m->m     m->i     m->d     i->m     i->i     d->m     d->d
  COMPO   2.36553  4.52577  2.96709  2.70473  3.20818  3.02239  3.41069  2.90041  2.55332  2.35210
          2.68640  4.42247  2.77497  2.73145  3.46376  2.40504  3.72516  3.29302  2.67763  2.69377
          0.57544  1.78073  1.31293  1.75577  0.18968  0.00000        *
      1   1.70038  4.17733  3.76164  3.36686  3.72281  3.29583  4.27570  2.40482  3.29230  2.54324
          2.68618  4.42225  2.77519  2.73123  3.46354  2.40513  3.72494  3.29354  2.67741  2.69355
          0.03156  3.86736  4.58970  0.61958  0.77255  0.34406  1.23405
...
    149   2.92198  5.11574  3.28049  2.65489  4.47826  3.59727  2.51142  3.88373  1.57593  3.35205
          2.68634  4.42241  2.77536  2.73098  3.46370  2.40469  3.72511  3.29370  2.67757  2.69331
          0.22163  1.61553        *  1.50361  0.25145  0.00000        *
//
```

HMMER

DOWNLOAD DOCUMENTATION SEARCH PUBLICATIONS BLOG

HMMER: biosequence analysis using profile hidden Markov models