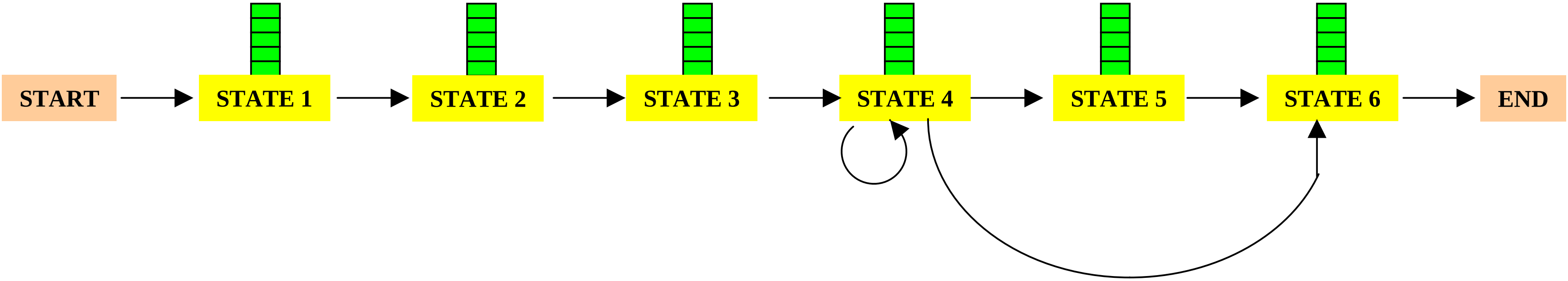
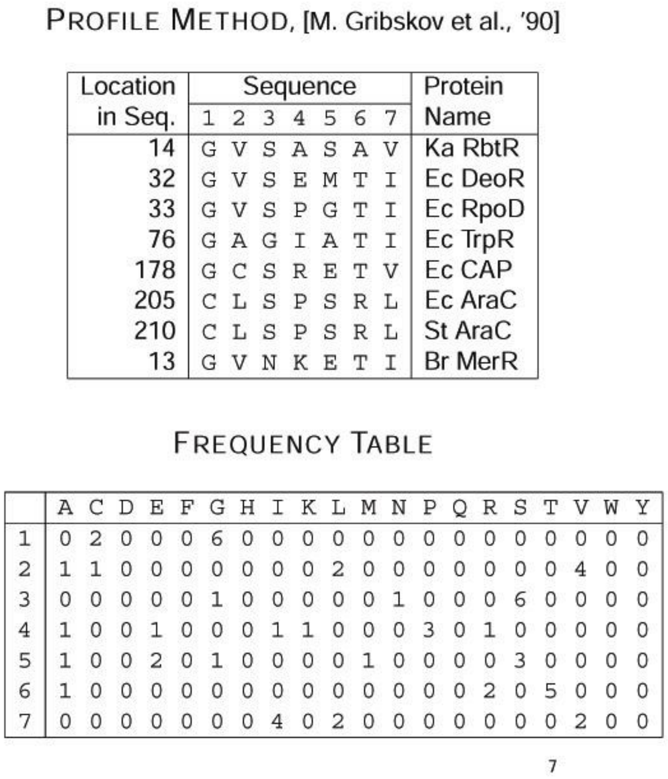
Simple Models

- Helps to model simple sequence features.
  - single sequences e.g. **TTGACA** or **TATATT** **[??]**
  - sets of sequences e.g. **[AT] C [GC] TC [AGC]**
  - sets of sequences with inserts e.g. **GCA [AT] [AT]* AG**
  - & deletes too, e.g. **TATA [G –] T**



  - long sequences with a sequence of domains **H-B-T-B-H**

Profile Method



PROFILE METHOD, [M. Gribskov et al., '90]

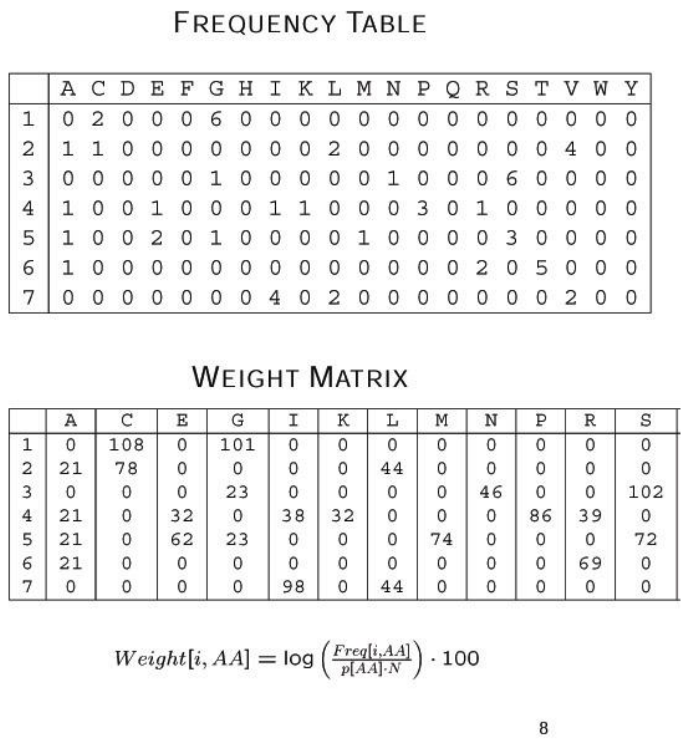
Location in Seq.	Sequence 1	2	3	4	5	6	7	Protein Name
14	G	V	S	A	S	A	V	Ka RbtR
32	G	V	S	E	M	T	I	Ec DeoR
33	G	V	S	P	G	T	I	Ec RpoD
76	G	A	G	I	A	T	I	Ec TrpR
178	G	C	S	R	E	T	V	Ec CAP
205	C	L	S	P	S	R	L	Ec AraC
210	C	L	S	P	S	R	L	St AraC
13	G	V	N	K	E	T	I	Br MerR

FREQUENCY TABLE

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

7

Profile Method



FREQUENCY TABLE

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

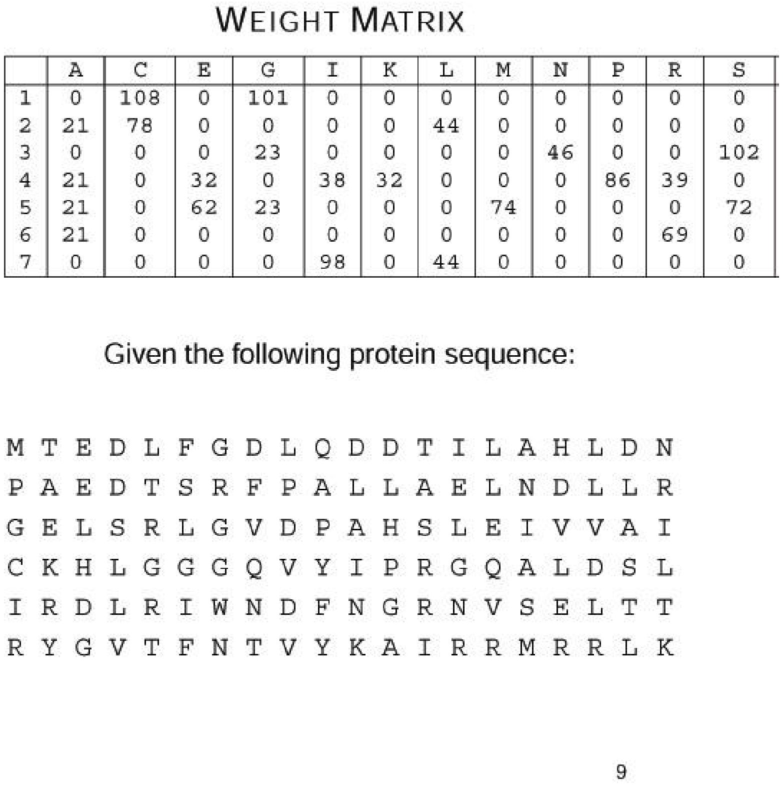
WEIGHT MATRIX

	A	C	E	G	I	K	L	M	N	P	R	S
1	0	108	0	101	0	0	0	0	0	0	0	0
2	21	78	0	0	0	0	44	0	0	0	0	0
3	0	0	0	23	0	0	0	0	46	0	0	102
4	21	0	32	0	38	32	0	0	0	86	39	0
5	21	0	62	23	0	0	0	74	0	0	0	72
6	21	0	0	0	0	0	0	0	0	0	69	0
7	0	0	0	0	98	0	44	0	0	0	0	0

$$Weight[i, AA] = \log\left(\frac{Freq[i,AA]}{p[AA]\cdot N}\right)\cdot 100$$

8

Profile Method



WEIGHT MATRIX

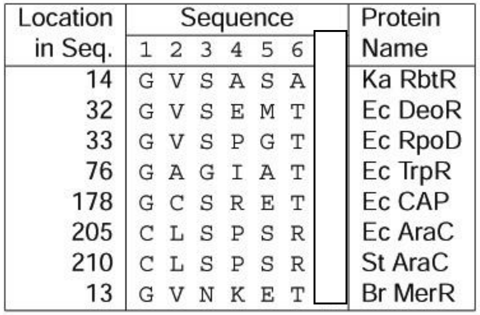
	A	C	E	G	I	K	L	M	N	P	R	S
1	0	108	0	101	0	0	0	0	0	0	0	0
2	21	78	0	0	0	0	44	0	0	0	0	0
3	0	0	0	23	0	0	0	0	46	0	0	102
4	21	0	32	0	38	32	0	0	0	86	39	0
5	21	0	62	23	0	0	0	74	0	0	0	72
6	21	0	0	0	0	0	0	0	0	0	69	0
7	0	0	0	0	98	0	44	0	0	0	0	0

Given the following protein sequence:

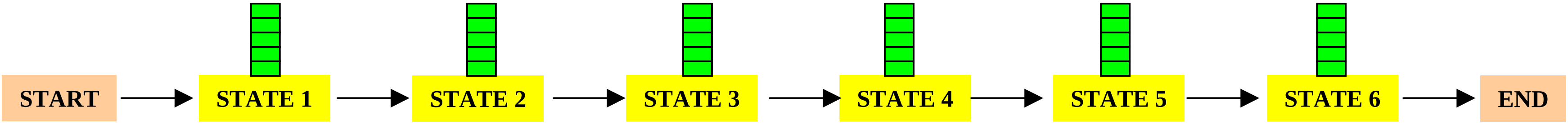
```
M T E D L F G D L Q D D T I L A H L D N
P A E D T S R F P A L L A E L N D L L R
G E L S R L G V D P A H S L E I V V A I
C K H L G G G Q V Y I P R G Q A L D S L
I R D L R I W N D F N G R N V S E L T T
R Y G V T F N T V Y K A I R R M R R L K
```

9

Profile HMMs



Location in Seq.	Sequence 1	2	3	4	5	6	Protein Name
14	G	V	S	A	S	A	Ka RbtR
32	G	V	S	E	M	T	Ec DeoR
33	G	V	S	P	G	T	Ec RpoD
76	G	A	G	I	A	T	Ec TrpR
178	G	C	S	R	E	T	Ec CAP
205	C	L	S	P	S	R	Ec AraC
210	C	L	S	P	S	R	St AraC
13	G	V	N	K	E	T	Br MerR



CpG Islands

- Regions in DNA sequences with increased occurrences of substring “CG”
- Rare: typically C gets methylated and then mutated into a T.
- Often around promoter or “start” regions of genes
- Few hundred to a few thousand bases long

Problem 1:

- Input: Small sequence S
- Output: Is S from a CpG island?
  - Build Markov Models: M+ & M-
  - Then compare

Problem 2:

- Input: Long sequence S
- Output: Identify the CpG islands in S.
  - Markov models are inadequate.
  - Need Hidden Markov Models.

Markov Models

+	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

How to distinguish?

- Compute

$$S(x) = \log\left(\frac{P(x \mid M+)}{P(x \mid M-)}\right) = \sum_{i=1}^{L} \log\left(\frac{p_{x(i-1)xi}}{m_{x(i-1)xi}}\right) = \sum_{i=1}^{L} r_{x(i-1)xi}$$

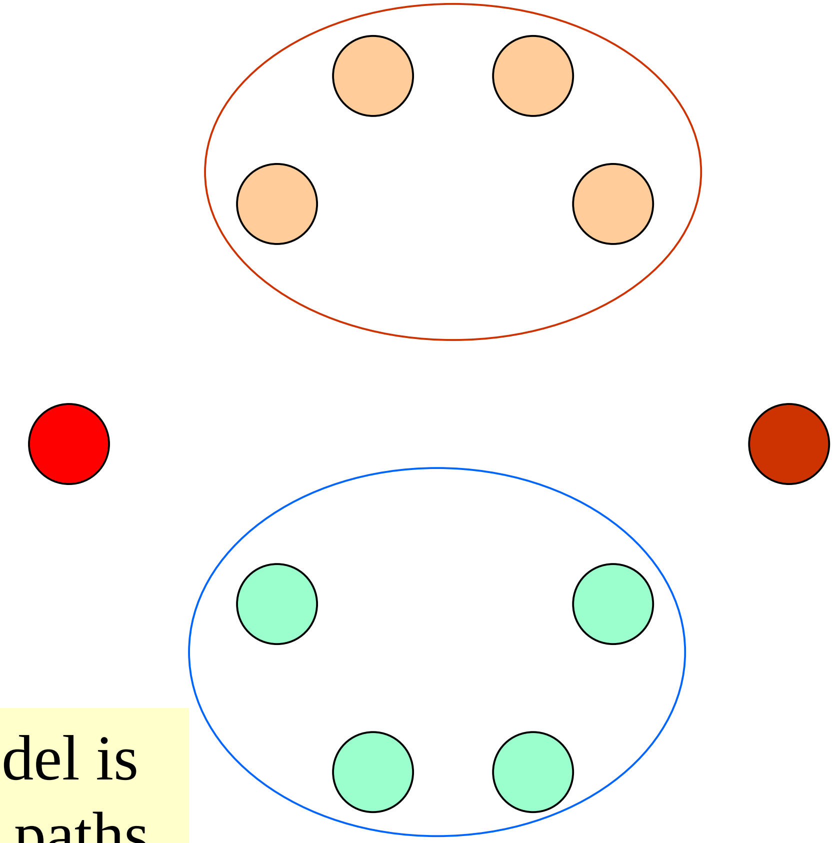
r=p/m	A	C	G	T
A	-0.740	0.419	0.580	-0.803
C	-0.913	0.302	1.812	-0.685
G	-0.624	0.461	0.331	-0.730
T	-1.169	0.573	0.393	-0.679

Score(GCAC)
= .461-.913+.419
< 0.
GCAC not from CpG island.

Score(GCTC)
= .461-.685+.573
> 0.
GCTC from CpG island.

Hidden Markov Model (HMM)

- States
- Transitions
- Transition Probabilities
- Emissions
- Emission Probabilities

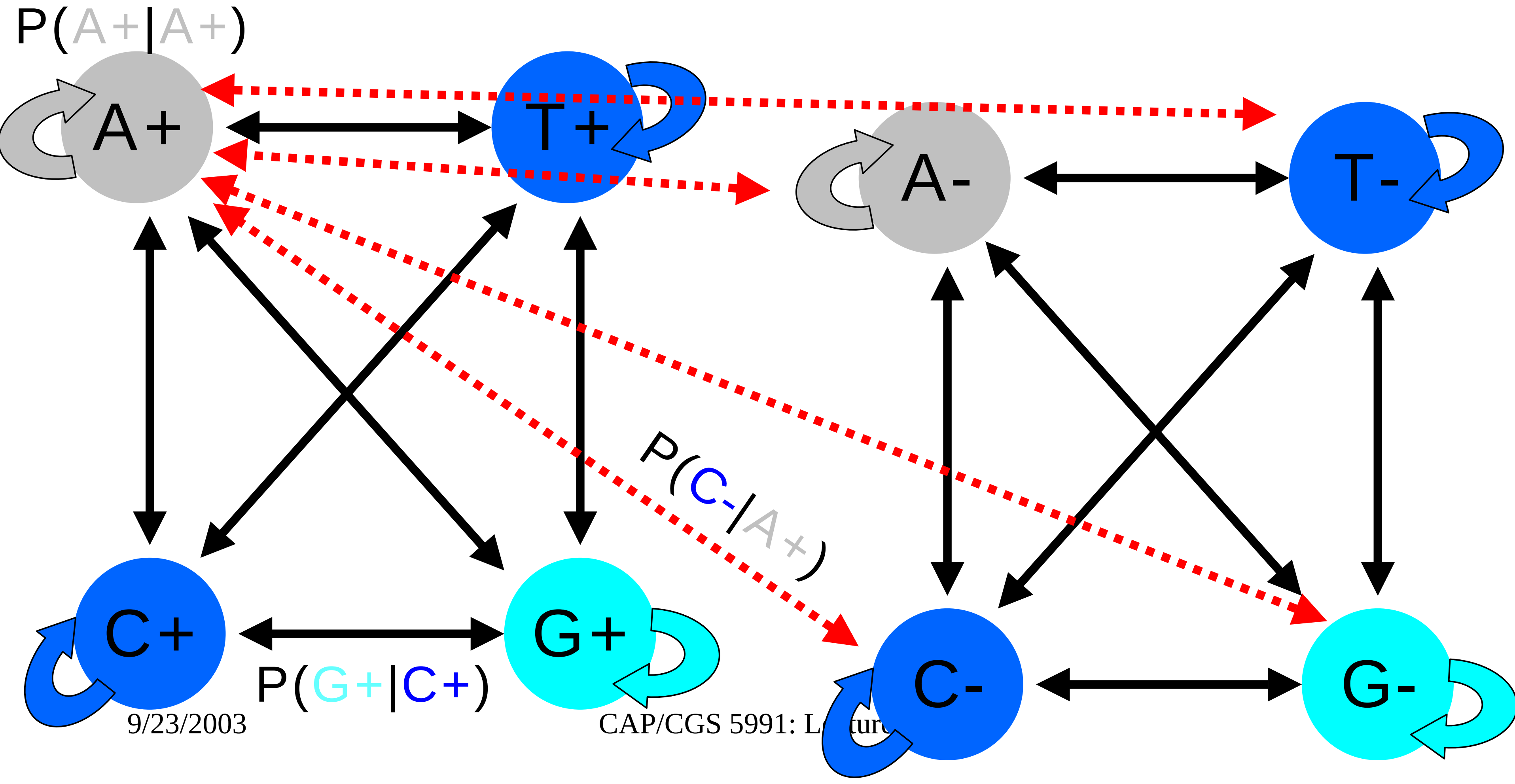


- What is hidden about HMMs?

Answer: The path through the model is hidden since there are many valid paths.

CpG Island + in an ocean of –
First order Hidden Markov Model

MM=16, HMM= 64 transition probabilities (adjacent bp)



How to Solve Problem 2?

- Solve the following problem:

Input: Hidden Markov Model M,

parameters Θ, emitted sequence S

Output: Most Probable Path Π

How: Viterbi's Algorithm (Dynamic Programming)

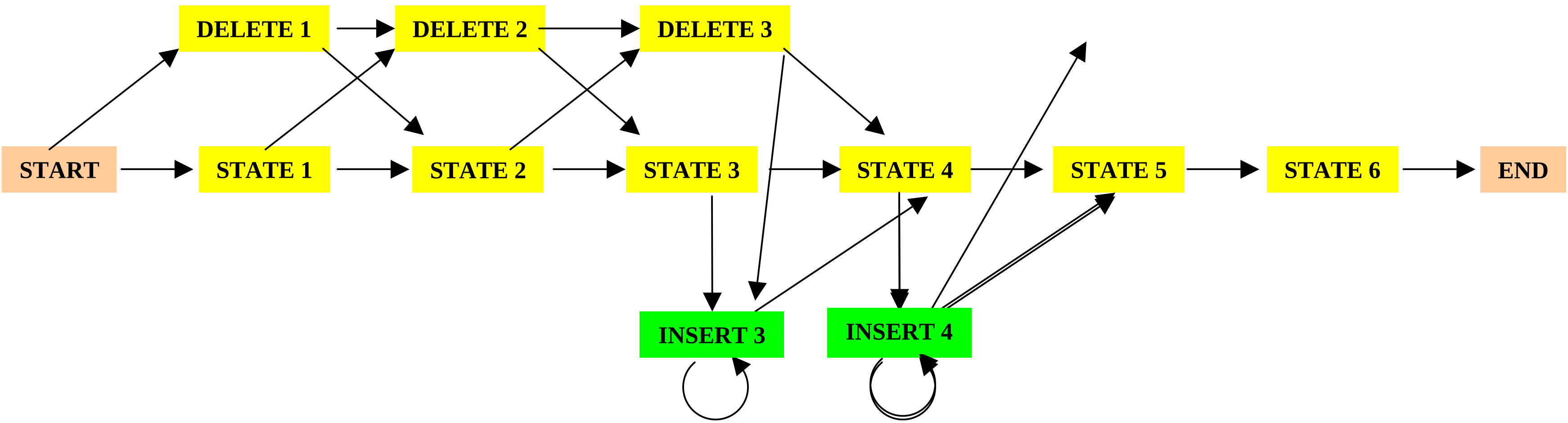
Define Π[i,j] = MPP for first j characters of S ending in state i

Define P[i,j] = Probability of Π[i,j]

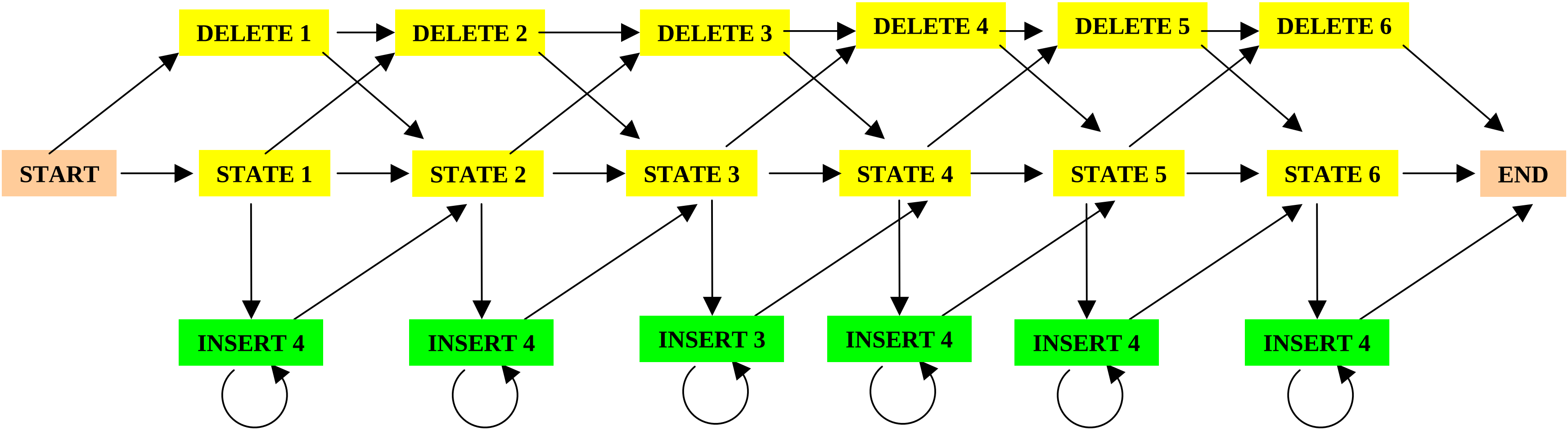
- Compute state i with largest P[i,j].

Profile HMMs with InDels

- Insertions
- Deletions
- Insertions & Deletions



Profile HMMs with InDels

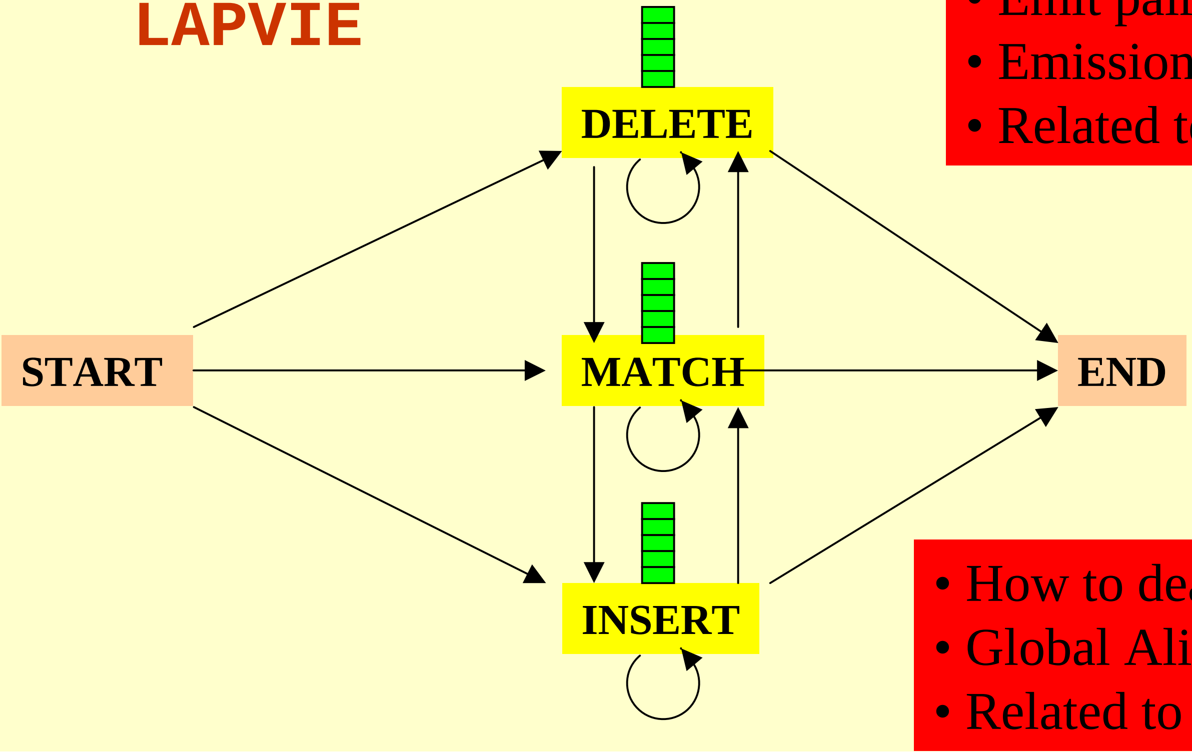


Missing transitions from DELETE j to INSERT j and
from INSERT j to DELETE j+1.

How to model Pairwise Sequence Alignment

LEAPVE

LAPVIE

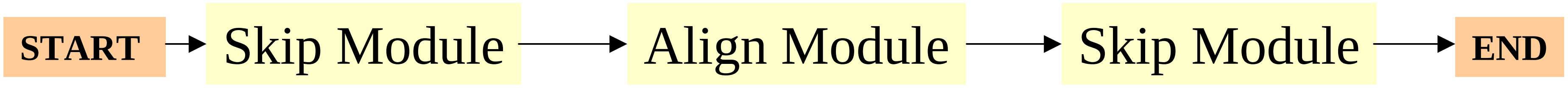


Pair HMMs

- Emit pairs of synbols
- Emission probs?
- Related to Sub. Matrices

- How to deal with InDels?
- Global Alignment? Local?
- Related to Sub. Matrices

How to model Pairwise Local Alignments?


START

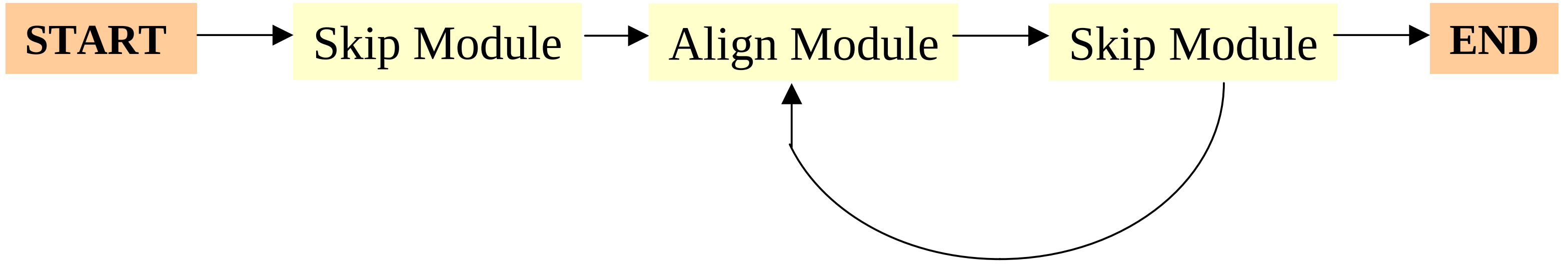
Skip Module

Align Module

Skip Module

END

How to model Pairwise Local Alignments with gaps?


START

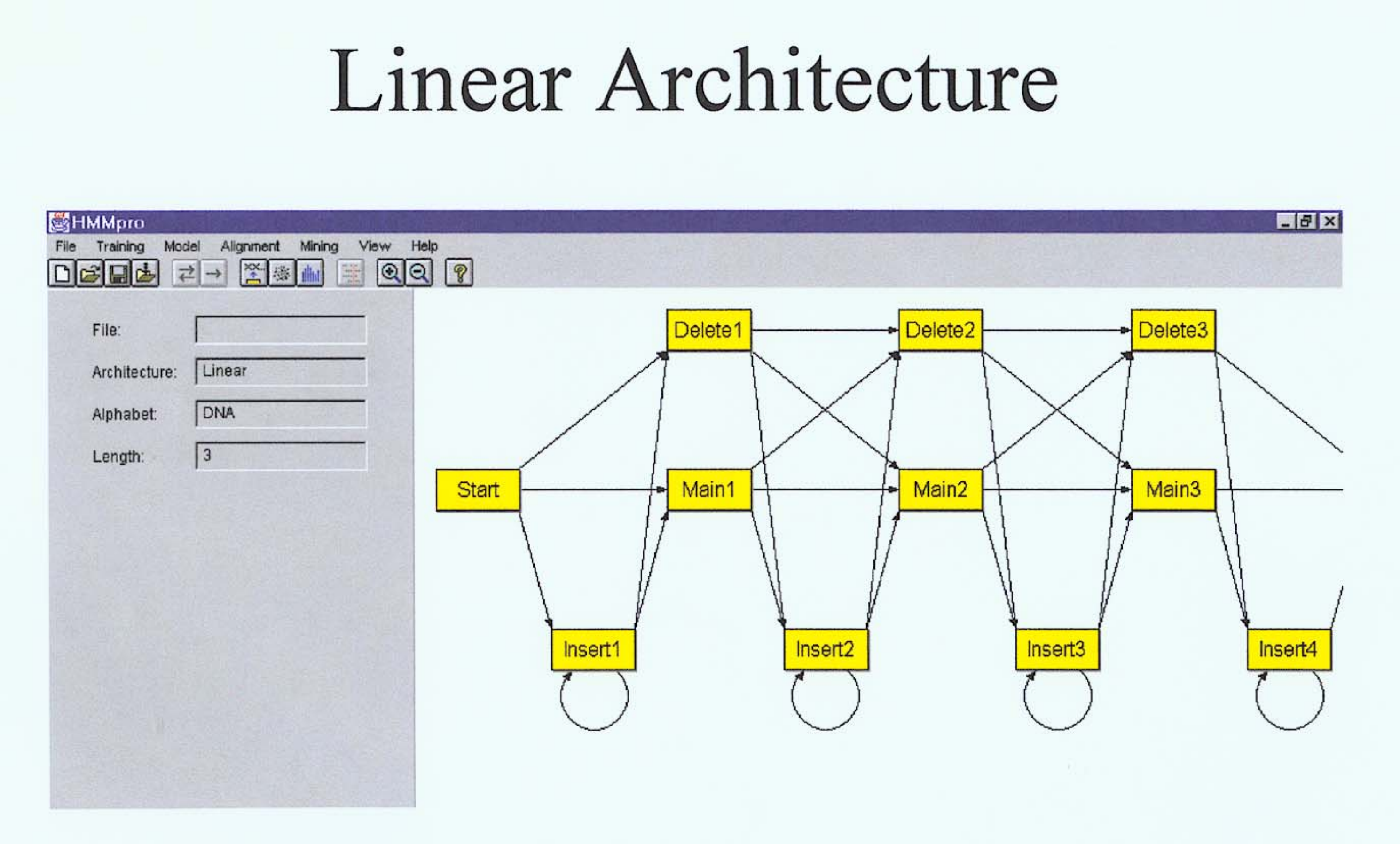
Skip Module

Align Module

Skip Module

END

Standard HMM architectures



Linear Architecture

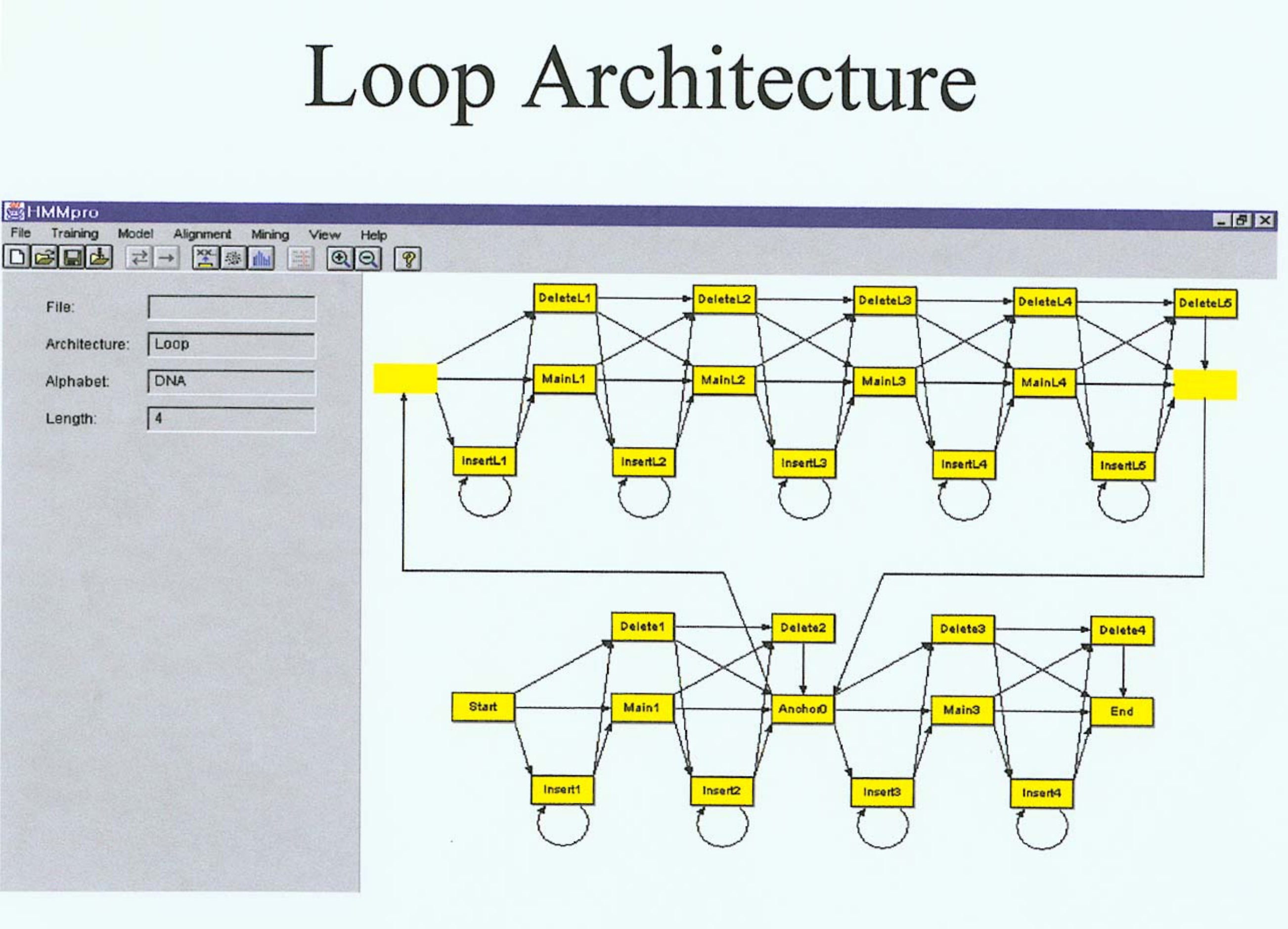
File:

Architecture: Linear

Alphabet: DNA

Length: 3

Standard HMM architectures



Loop Architecture

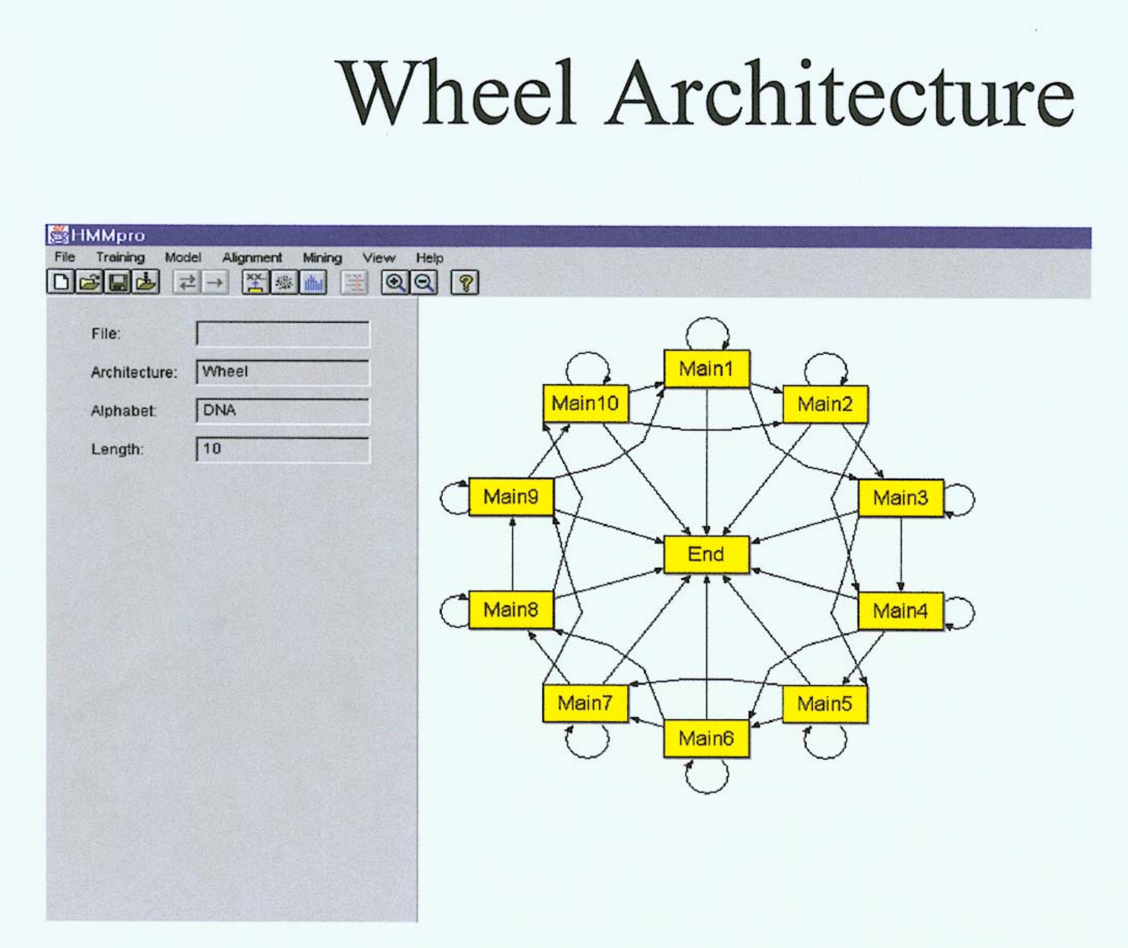
File:

Architecture: Loop

Alphabet: DNA

Length: 4

Standard HMM architectures



Wheel Architecture

File:

Architecture: Wheel

Alphabet: DNA

Length: 10

Profile HMMs from Multiple Alignments

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

Construct Profile HMM from above multiple alignment.

Problem 3: LIKELIHOOD QUESTION

- Input: Sequence S, model M, state i
- Output: Compute the probability of reaching state i with sequence S using model M
  - Backward Algorithm (DP)

Problem 4: LIKELIHOOD QUESTION

- Input: Sequence S, model M
- Output: Compute the probability that S was emitted by model M
  - Forward Algorithm (DP)

Problem 5: LEARNING QUESTION

- Input: model structure M, Training Sequence *S*
- Output: Compute the parameters Θ
- Criteria: ML criterion
  - maximize P(*S* | M, Θ)   HOW???

Problem 6: DESIGN QUESTION

- Input: Training Sequence *S*
- Output: Choose model structure M, and compute the parameters Θ
  - No reasonable solution
  - Standard models to pick from

Iterative Solution to the LEARNING QUESTION (Problem 5)

- Pick initial values for parameters Θ_0
- Repeat

```
        Run training set $S$ on model M
        Count # of times transition i $\Rightarrow$ j is made
        Count # of times letter x is emitted from state i
        Update parameters $\Theta$
```

- Until (some stopping condition)

Entropy

- Entropy measures the variability observed in given data.

$$E = -\sum_{c} p_c \log p_c$$

- Entropy is useful in multiple alignments & profiles.
- Entropy is max when uncertainty is max.

G-Protein Couple Receptors

- Transmembrane proteins with 7 α-helices and 6 loops; many subfamilies
- Highly variable: 200-1200 aa in length, some have only 20% identity.
- [Baldi & Chauvin, ’94] HMM for GPCRs
- HMM constructed with 430 match states (avg length of sequences) ; Training: with 142 sequences, 12 iterations

GPCR - Analysis

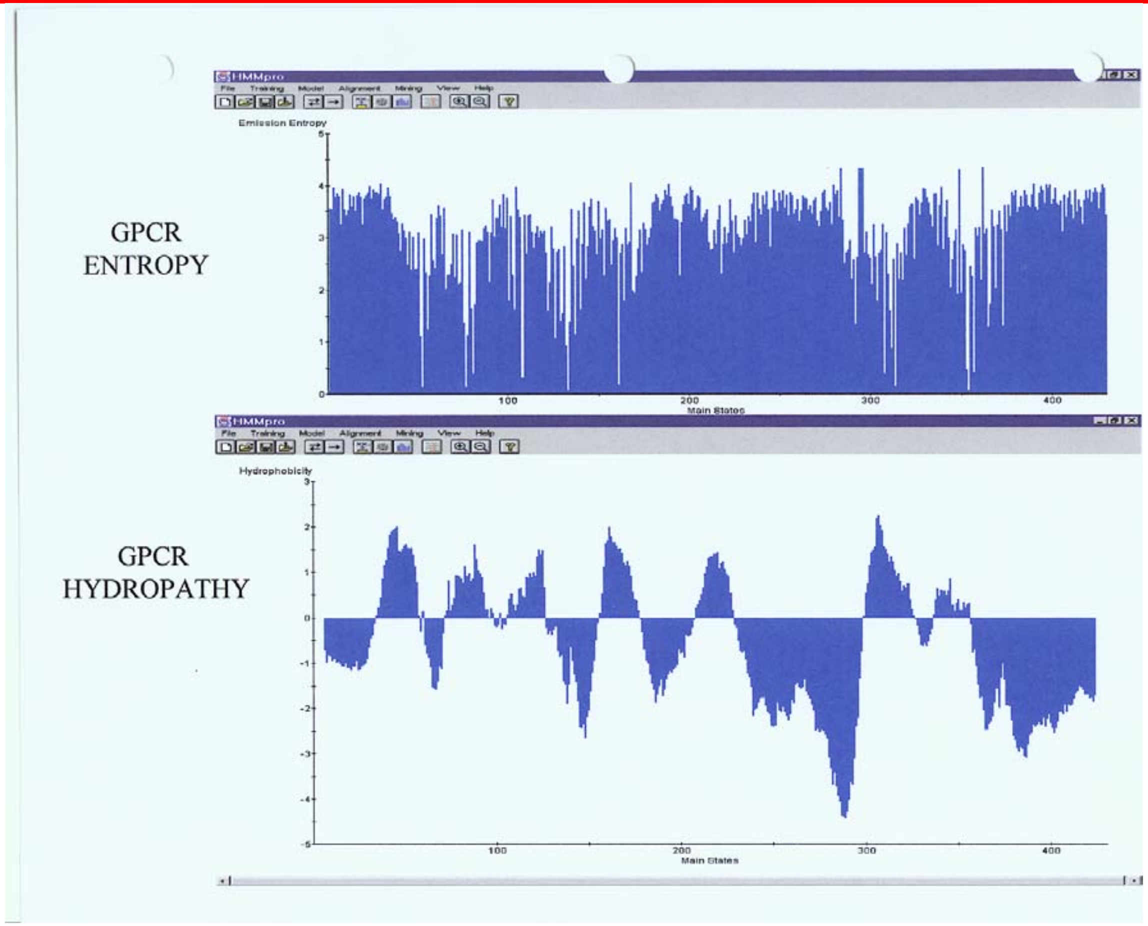
- Compute main state entropy values

$$H_i = -\sum_a e_{ia} \log e_{ia}$$

- For every sequence from test set (142) & random set (1600) & all SWISS-PROT proteins
  - Compute the negative log of probability of the most probable path π

$$Score(S) = -\log\left(P(\pi \mid S, M)\right)$$

GPCR Analysis



Entropy

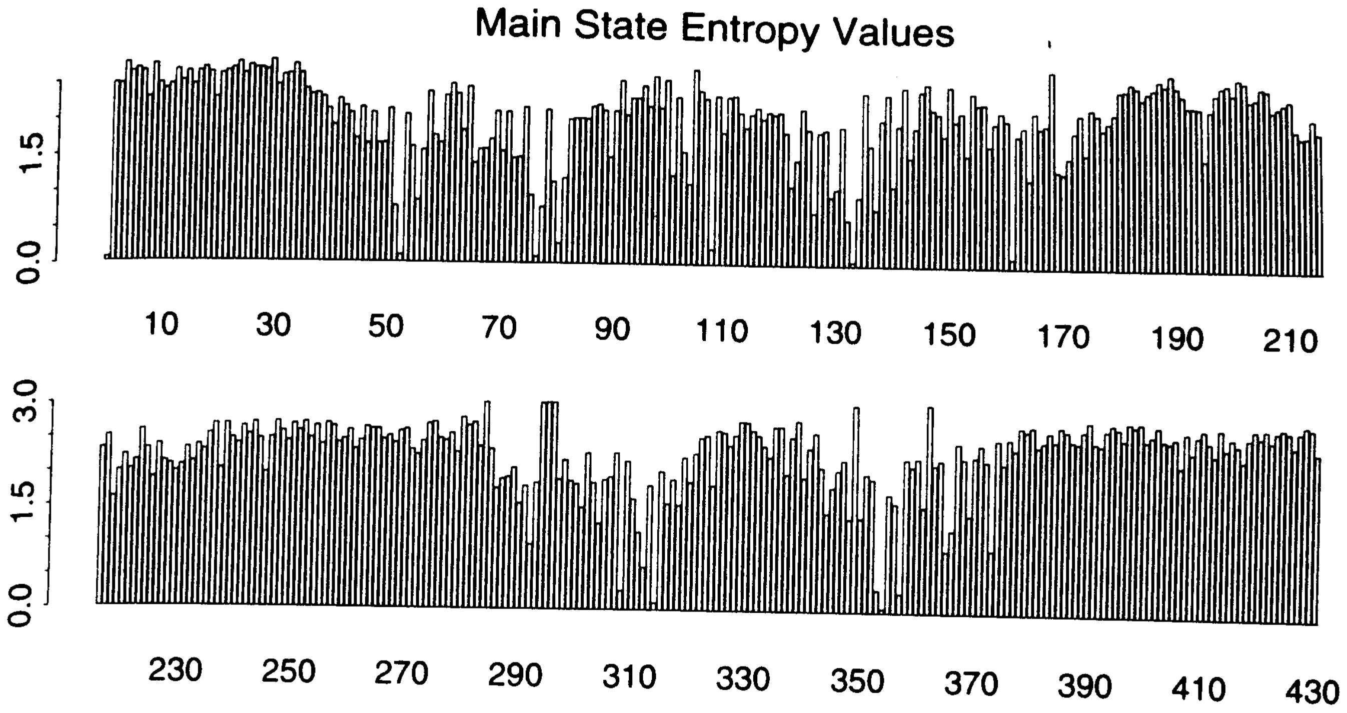


Figure 8.1: Entropy Profile of the Emission Probability Distributions Associated with the Main States of the HMM After 12 Cycles of Training.

GPCR Analysis (Cont’d)

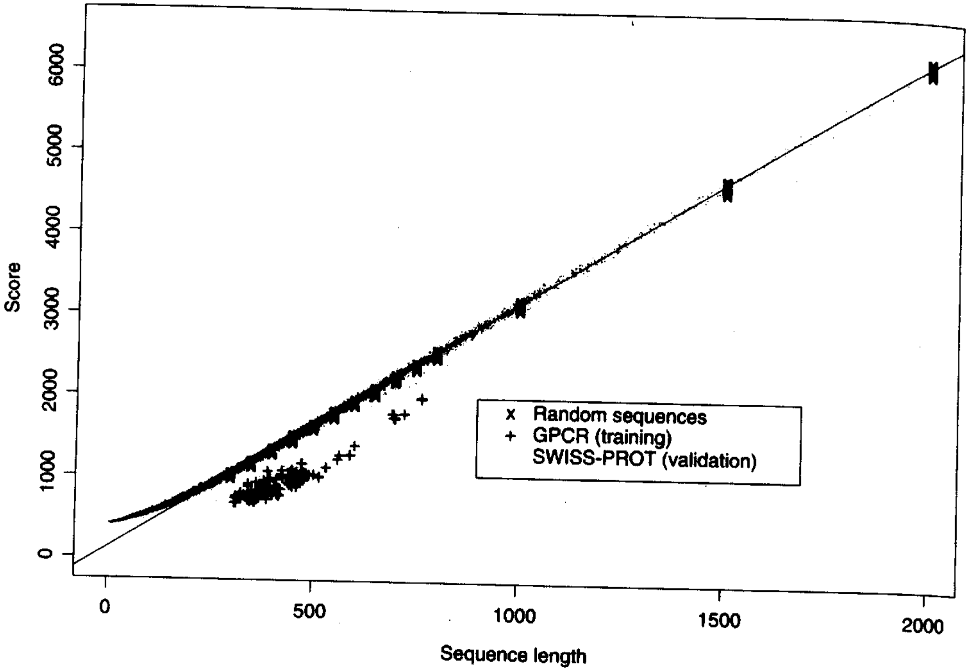


Figure 8.2: Scores (Negative Log-likelihoods of Optimal Viterbi Paths). Represented sequences consist of 142 GPCR training sequences, all sequences from the SWISS-PROT database of length less than or equal to 2000, and 220 randomly generated sequences with same average composition as the GPCRs of length 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800 (20 at each length). The regression line was obtained from the 220 random sequences. The horizontal distances in the histogram correspond to(malized scores (6).

Applications of HMM for GPCR

- Bacteriorhodopsin
  - Transmembrane protein with 7 domains
  - But it is not a GPCR
  - Compute score and discover that it is close to the regression line. Hence not a GPCR.
- Thyrotropin receptor precursors
  - All have long initial loop on INSERT STATE 20.
  - Also clustering possible based on distance to regression line.

HMMs – Advantages

- Sound statistical foundations
- Efficient learning algorithms
- Consistent treatment for insert/delete penalties for alignments in the form of locally learnable probabilities
- Capable of handling inputs of variable length
- Can be built in a modular & hierarchical fashion; can be combined into libraries.
- Wide variety of applications: Multiple Alignment, Data mining & classification, Structural Analysis, Pattern discovery, Gene prediction.

HMMs – Disadvantages

- Large # of parameters.
- Cannot express dependencies & correlations between hidden states.

Shift-And Method (Baeza-Yates & Gonnet)

Idea: *Build a bit-matrix M* such that

M[I,J] = 1 ⇔ P[1..I] = T[J-I+1 .. J]

Thus, M[I,J] = 1 ⇔ (M[I-1, J-1] = 1) & (P[I] = T[J])

M	T	A	G	T	A	G	A	A	G	A	A	C
A	0	1	0	0	1	0	1	1	0	1	1	0
G		0	1	0	0	1	0	0	1	0	0	0
A			0	0	0	0	1	0	0	1	0	0
A				0	0	0	0	1	0	0	1	0
C					0	0	0	0	0	0	0	1

Shift-And (Cont'd)

Idea: *Operate on column bit-vectors*

Step 1: Build a bit-matrix U such that for each $e \in \Sigma$

$U[I,e] = 1 \Leftrightarrow P[I] = e$

U	A	C	G	T
A	1	0	0	0
G	0	0	1	0
A	1	0	0	0
A	1	0	0	0
C	0	1	0	0

Step 2: M[J] = RightShift(M[J-1]) && U[T[J]]

Shift-And (Cont’d)

Step 2: M[J] = RightShift(M[J-1]) && U[T[J]]

M	T	A	G	T	A	G	A	A	G	A	A	C
A	0	1	0	0	1	0	1	1	0	1	1	0
G		0	1	0	0	1	0	0	1	0	0	0
A			0	0	0	0	1	0	0	1	0	0
A				0	0	0	0	1	0	0	1	0
C					0	0	0	0	0	0	0	1

U	A	C	G	T
A	1	0	0	0
G	0	0	1	0
A	1	0	0	0
A	1	0	0	0
C	0	1	0	0

Shift-And (Generalizations)

Generalization 1: *Wild Cards*: match all characters.

U	A	C	G	T
A	1	0	0	0
G	0	0	1	0
A	1	0	0	0
*	1	1	1	1
C	0	1	0	0

Generalization 2: *k Mismatches*: Compute M_0, M_1, …, M_k

$M_s[J]$ = RightShift($M_{s-1}[J-1]$ **AND** $U[T[J]]$)
OR $M_{s-1}[J]$
OR $M_{s-1}[J-1]$

String Matching Methods: Overview

Methods:

- Naïve Method **O(mn)** *time*
- Rabin Karp Method **O(mn)** *time;* Fast on average.
- FSA-based method **O(n+mA)** *time*
- Knuth-Morris-Pratt algorithm **O(n+m)** *time*
- Boyer-Moore **O(mn)** time; Very fast on average.
- Suffix Tree method; **O(m+n)** *time*
- Shift-And method; Fast on average; Bit operations.