

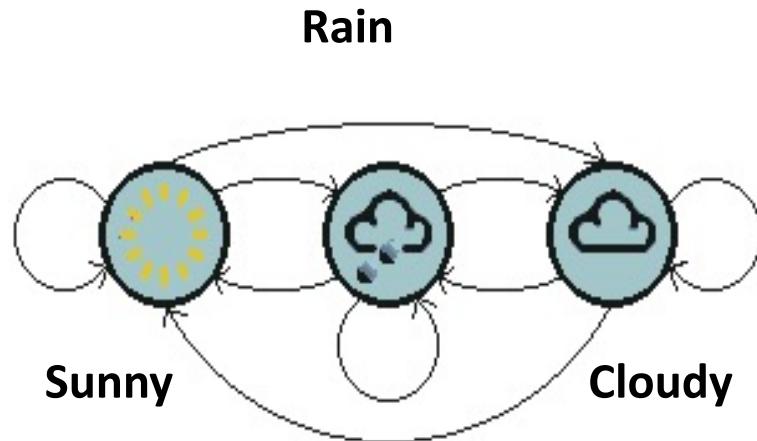
Hidden Markov Models, HMM's

Morten Nielsen,
Department of Health
Technology DTU
and
Instituto de Investigaciones
Biotecnológicas, Universidad
Nacional de San Martín,
Argentina

Objectives

- Introduce Hidden Markov models and understand that they are just weight matrices with gaps
- How to construct an HMM
- How to “align/score” sequences to HMM’s
 - Viterbi decoding
- Use and construct Profile HMM
 - HMMer

Markov Chains



		weather today			
		Sun	Cloud	Rain	
weather yesterday	Sun	0.5	0.25	0.25	
	Cloud	0.375	0.125	0.375	
		Rain	0.125	0.625	0.375

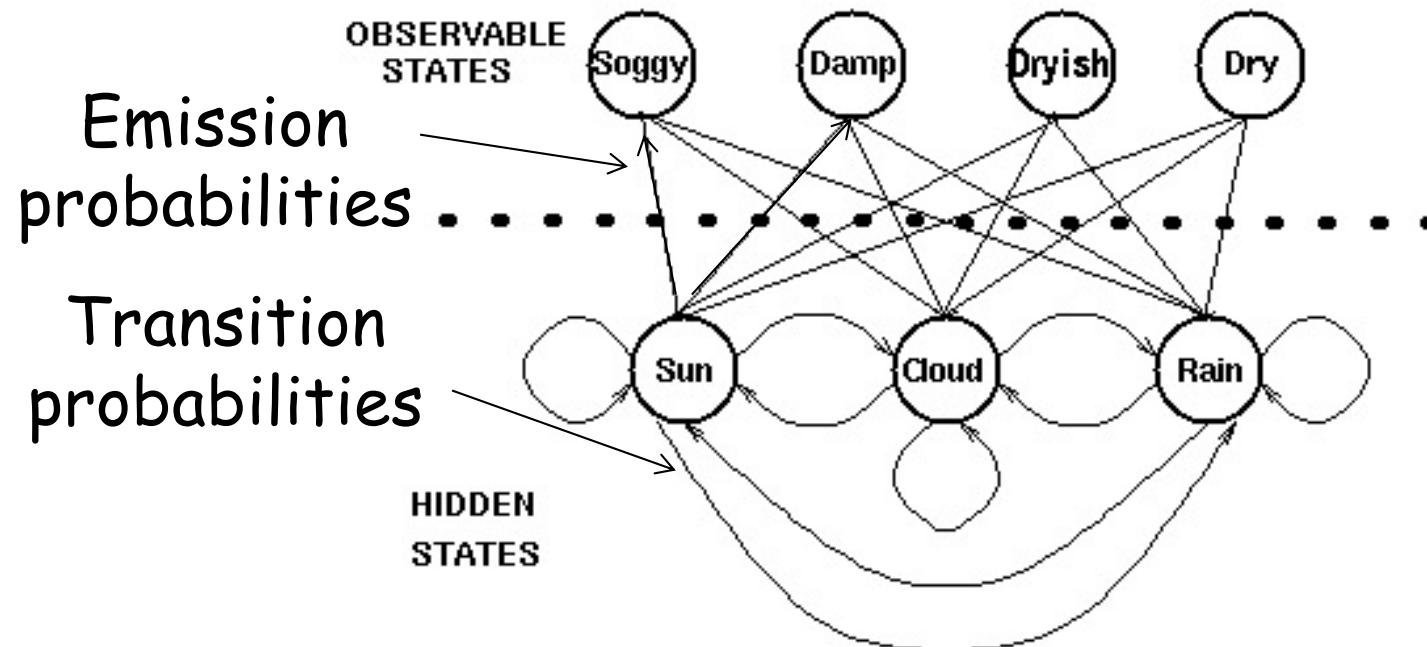
States : Three states - sunny, cloudy, rainy.

State transition matrix : The probability of the weather given the previous day's weather.

$$\begin{pmatrix} \text{Sun} & \text{Cloud} & \text{Rain} \\ 1.0 & 0.0 & 0.0 \end{pmatrix}$$

Initial Distribution : Defining the probability of the system being in each of the states at time 0.

Hidden Markov Models



Hidden states : the (TRUE) states of a system that may be described by a Markov process (e.g., the weather).

Observable states : the states of the process that are 'visible' (e.g., seaweed dampness).

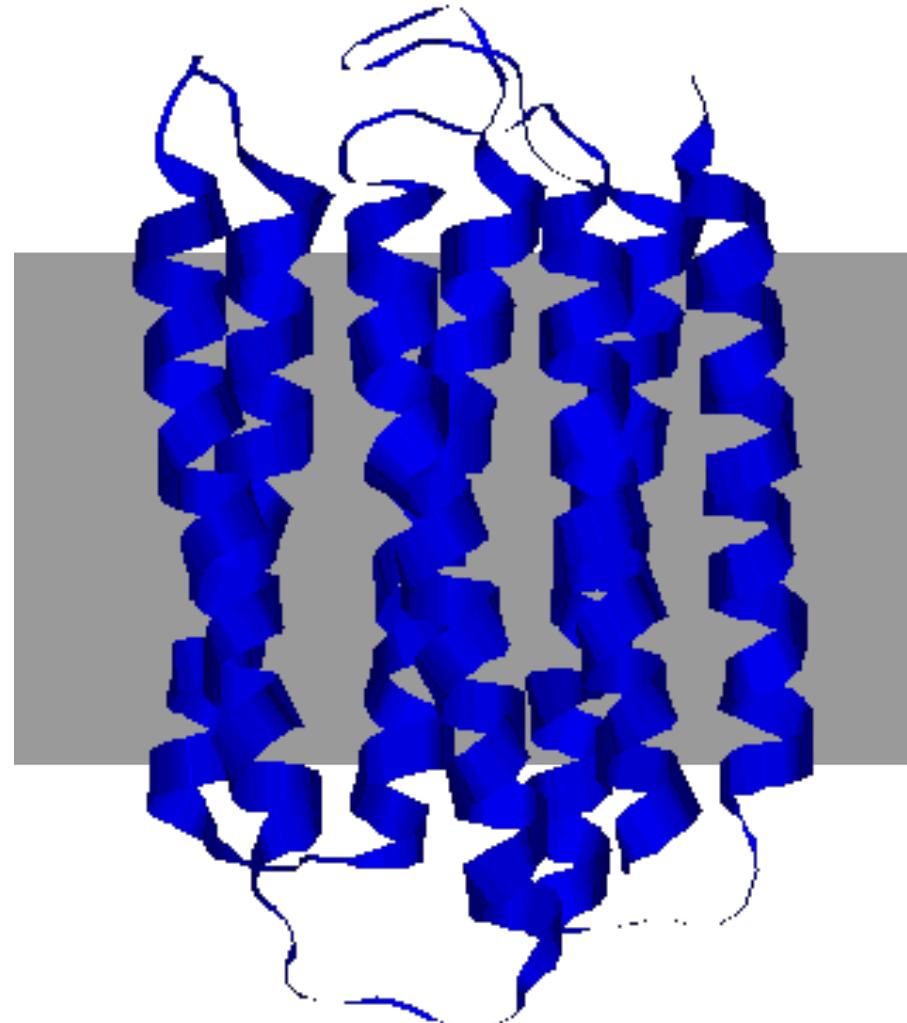
TMHMM (trans-membrane HMM) (Sonnhammer, von Heijne, and Krogh)

CENTERFO
RBIOLOGI
CALSEQU
ENCEANA
LYSIS CBS

Extra cellular

Trans membrane

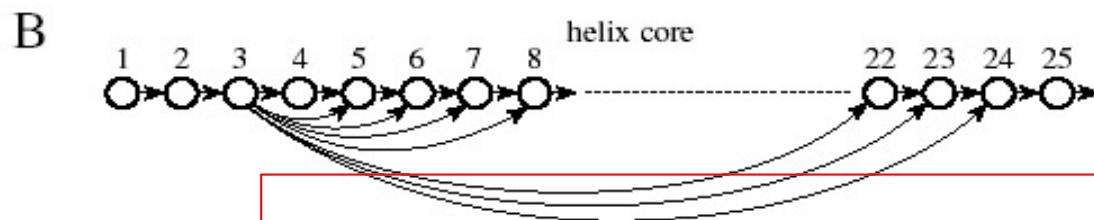
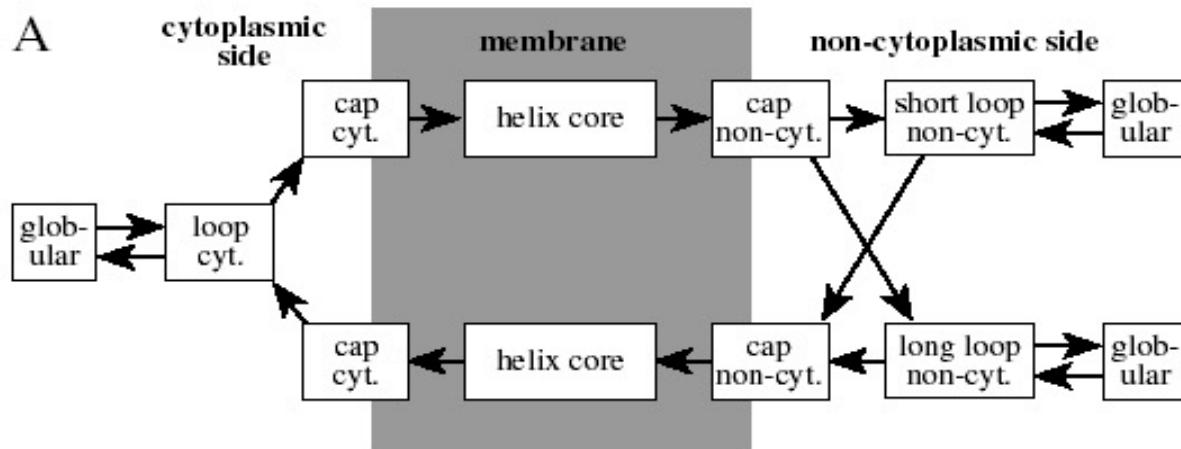
Intra cellular



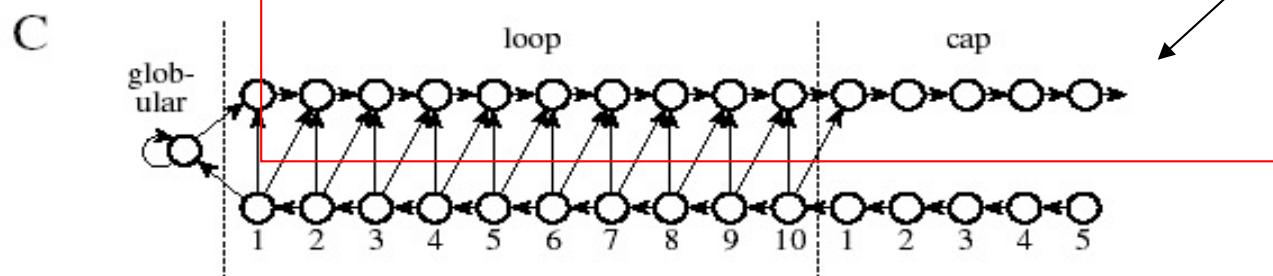
TMHMM (trans-membrane HMM)

(Sonnhammer, von Heijne, and Krogh)

CENTER FOR BIOLOGICAL
SEQUENCE ANALYSIS CBS



Model TM length distribution.
Power of HMM.
Difficult in alignment.



ALLYVDWQILPVIL

Weight matrix construction

SLLPAIVEL YLLPAIVHI TLWVDPYEV GLVPFLVSV KLLEPVLLL LLDVPTAAV LLDVPTAAV LLDVPTAAV
LLDVPTAAV VLFRGGPRG MVDGTLLLL YMNGTMSQV MLLSVPLLL SLLGLLVEV ALLPPINIL TLIKIQHTL
HLIDYLVTS ILAPPVVKL ALFPQLVIL GILGFVFTL STNRQSGRQ GLDVLTAKV RILGAVAKV QVCERIPTI
ILFGHENRV ILMEMHIHKL ILDQKINEV SLAGGIIGV LLIENVASL FLLWATAEA SLPDFGISY KKREEAPSL
LERPGGNEI ALSNLEVKL ALNELLQHV DLERKVESL FLGENISNF ALSDHHIYL GLSEFTEYL STAPPAHGV
PLDGEYFTL GVLVGVALI RTLDKVLEV HLSTAFARV RLDSYVRSL YMNGTMSQV GILGFVFTL ILKEPVHGV
ILGFVFTLT LLFGYPVYV GLSPPTVWLS WLSLLVPFV FLPSDFFPS CLGGLLTMV FIAGNSAYE KLGEFYNM
KLVALGINA DLMGYIPLV RLVTLKDIV MLLAVLYCL AAGIGILT YLEPGPVTA LLDGTATLR ITDQVPFSV
KTWGQYWQV TITDQVPFS AFHHVAREL YLNKIQNSL MMRKLAILS AIMDKNIIL IMDKNIILK SMVGNWAKV
SLLAPGAKQ KIFGSLAFL ELVSEFSRM KLTPLCVTL VLYRYGSFS YIGEVLVSV CINGVCWTV VMNILLQYV
ILTVILGVL KVLEYVIKV FLWGPRALV GLSRVYARL FLLTRILTI HLGNVKYLV GIAGGLALL GLQDCTMLV
TGAPVTYST VIYQYMDDL VLPDVFIRC VLPDVFIRC AVGIGIAVV LVVLGLLAV ALGLGLLPV GIGIGVLA
GAGIGVAVL IAGIGILAI LIVIGILIL LAGIGLIAA VDGIGILTI GAGIGVLTA AAGIGIIQI QAGIGILLA
KARDPHSGH KACDPHSGH ACDPHSGHF SLYNTVATL RGPGRAFVT NLVPMVATV GLHCYEQLV PLKQHFQIV
AVFDRKSDA LLDFVRFMG VLVKSPNHW GLAPPQHLLI LLGRNSFEV PLTFGWCYK VLEWRFDSR TLNAWVKVV
GLCTLVAML FIDSYICQV IISAVVGIL VMAGVGSPY LLWTLVVLL SVRDRLARL LLMDCSGSI CLTSTVQLV
VLHDDILLEA LMWITQCFL SLLMWITQC QLSLLMWIT LLGATCMFV RLTRFLSRV YMDGTMQV FLTPKKLQC
ISNDVCAQV VKTDGNPPE SVYDFFVWL FLYGALLA VLFSSDFRI LMWAKIGPV SLLLELEEV SLSRFWSWA
YTAFTIPSI RLMKQDFSV RLPRTFCSC FLWGPRAYA RLLQETELV SLFEGIDFY SLDQSVVEL RLNMFTPYI
NMFTPYIGV LMI IPLINV TLFIGSHVV SLVIVTTFV VLQWASLAV ILAKFLHWL STAPPHVNV LLLLTVLTV
VVLGVVFGI ILHNGAYSL MIMVKCWMI MLGHTMEV MLGHTMEV SLADNSLA LLWAARPRL GVALQTMQ
GLYDGMEHL KMVELVHFL YLQLVFGIE MLMAQEALA LMAQEALAF VYDGREHTV YLSGANLNL RMFPNAPYL
EAAGIGILT TLDSQVMSL STPPPGRTRV KVAELVHFL IMIGVLGVV ALCRWGLLL LLFAGVQCQ VLLCESTAV
YLSTAFARV YLLEMILWRL SLDDYNHILV RTLDKVLEV GLPVEYLQV KLIANNTRV FIYAGSLSA KLVANNTRL
FLDEFMEGV ALQP GTALL VLDGLDVLL SLYSFPEPE ALYVDSLFF SLLQHLLIGL ELTLGEFLK MINAYLDKL
AAGIGILT V FLPSDFFPS SVRDRLARL SLREWLLRI LLSAWILTA AAGIGILT AVPDEIPPL FAYDGKDYI
AAGIGILT V FLPSDFFPS AAGIGILT V FLPSDFFPS AAGIGILT V FLWGPRALV ETVSEQSNV ITLWQRPLV

PSSM construction

- Calculate amino acid frequencies at each position using
 - Sequence weighting
 - Pseudo counts
- Define background model
 - Use background amino acids frequencies
- PSSM is

$$S(a_i) = \log \frac{p(a_i)}{q(a)}$$

More on scoring

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1	0.6	0.4	-3.5	-2.4	-0.4	-1.9	-2.7	0.3	-1.1	1.0	0.3	0.0	1.4	1.2	-2.7	1.4	-1.2	-2.0	1.1	0.7
2	-1.6	-6.6	-6.5	-5.4	-2.5	-4.0	-4.7	-3.7	-6.3	1.0	5.1	-3.7	3.1	-4.2	-4.3	-4.2	-0.2	-5.9	-3.8	0.4
3	0.2	-1.3	0.1	1.5	0.0	-1.8	-3.3	0.4	0.5	-1.0	0.3	-2.5	1.2	1.0	-0.1	-0.3	-0.5	3.4	1.6	0.0
4	-0.1	-0.1	-2.0	2.0	-1.6	0.5	0.8	2.0	-3.3	0.1	-1.7	-1.0	-2.2	-1.6	1.7	-0.6	-0.2	1.3	-6.8	-0.7
5	-1.6	-0.1	0.1	-2.2	-1.2	0.4	-0.5	1.9	1.2	-2.2	-0.5	-1.3	-2.2	1.7	1.2	-2.5	-0.1	1.7	1.5	1.0
6	-0.7	-1.4	-1.0	-2.3	1.1	-1.3	-1.4	-0.2	-1.0	1.8	0.8	-1.9	0.2	1.0	-0.4	-0.6	0.4	-0.5	-0.0	2.1
7	1.1	-3.8	-0.2	-1.3	1.3	-0.3	-1.3	-1.4	2.1	0.6	0.7	-5.0	1.1	0.9	1.3	-0.5	-0.9	2.9	-0.4	0.5
8	-2.2	1.0	-0.8	-2.9	-1.4	0.4	0.1	-0.4	0.2	-0.0	1.1	-0.5	-0.5	0.7	-0.3	0.8	0.8	-0.7	1.3	-1.1
9	-0.2	-3.5	-6.1	-4.5	0.7	-0.8	-2.5	-4.0	-2.6	0.9	2.8	-3.0	-1.8	-1.4	-6.2	-1.9	-1.6	-4.9	-1.6	4.5

$$S = \sum_i S(a_i)$$

$$S = \sum_i \log \frac{p(a_i)}{q(a_i)}$$

$$S = \log \left(\frac{\prod_i p(a_i)}{\prod_i q(a_i)} \right)$$

Probability of observation given Model

Probability of observation given Prior
(background)

$$S = \log \left(\frac{P(a | M)}{P(a | B)} \right)$$

Hidden Markov Models

- Weight matrices do not deal with insertions and deletions
- In alignments, this is done in an ad-hoc manner by optimization of the two gap penalties for first gap and gap extension
- HMM is a natural framework where insertions/deletions are dealt with explicitly

Multiple sequence alignment

Learning from evolution

QUERY	HAMDIRCYHSGG	PLHL	GEI	EDF	NGQSCIVCPWHKYKITLATGE	GLY	QSINPKDPS
Q8K2P6	HAMDIRCYHSGG	PLHL	GEI	EDF	NGQSCIVCPWHKYKITLATGE	GLY	QSINPKDPS
Q8TAC1	HAMDIRCYHSGG	PLHL	GDI	EDF	DGRPCIVCPWHKYKITLATGE	GLY	QSINPKDPS
Q07947	FAVQDTCTHGDW	ALSE	GYI	DGD	--VVECTLHFGKFCVRTGK--	VKAL	PA
P0A185	YATDNLCTHGSA	RMSD	GYI	EGRE	-IECPLHQGRFDVCTGK-	ALC	APV
P0A186	YATDNLCTHGSA	RMSD	GYI	EGRE	-IECPLHQGRFDVCTGK-	ALC	APV
Q51493	YATDNLCTHGAA	RMSD	GFI	EGRE	-IECPLHQGRFDVCTGR-	ALC	APV
A5W4F0	FAVQDTCTHGDW	ALSD	GYI	DGD	-IVECTLHFGKFCVRTGK-	VKAL	PA
P0C620	FAVQDTCTHGDW	ALSD	GYI	DGD	-IVECTLHFGKFCVRTGK-	VKAL	PA
P08086	FAVQDTCTHGDW	ALSD	GYI	DGD	-IVECTLHFGKFCVRTGK-	VKAL	PA
Q52440	FATQDQCTHGEW	SLSE	GGY	LDGD	-VVECSLHMGKFCVRTGK-	--	V
Q7N4V8	FAVDDRCSHGNA	SISE	GYI	ED	-NATVECPLHTASFCLRTGK-	ALCL	PA
P37332	FATQDRCTHGDW	SLSDG	GYI	EGD	-VVECSLHMGKFCVRTGK-	--	V
A7ZPY3	YAINDRCSHGNA	SMSE	GYI	EDD	-ATVECPLHAASFCLKTGK-	ALCL	PA
P0ABW1	YAINDRCSHGNA	SMSE	GYI	EDD	-ATVECPLHAASFCLKTGK-	ALCL	PA
A8A346	YAINDRCSHGNA	SMSE	GYI	EDD	-ATVECPLHAASFCLKTGK-	ALCL	PA
P0ABW0	YAINDRCSHGNA	SMSE	GYI	EDD	-ATVECPLHAASFCLKTGK-	ALCL	PA
P0ABW2	YAINDRCSHGNA	SMSE	GYI	EDD	-ATVECPLHAASFCLKTGK-	ALCL	PA
Q3YZ13	YAINDRCSHGNA	SMSE	GYI	EDD	-ATVECPLHAASFCLKTGK-	ALCL	PA
Q06458	YALDNLEPGSEANVLSR	GLI	GDAGGEPIVISPLYKQRIRLRDG	--	--	--	--



Core



Insertion

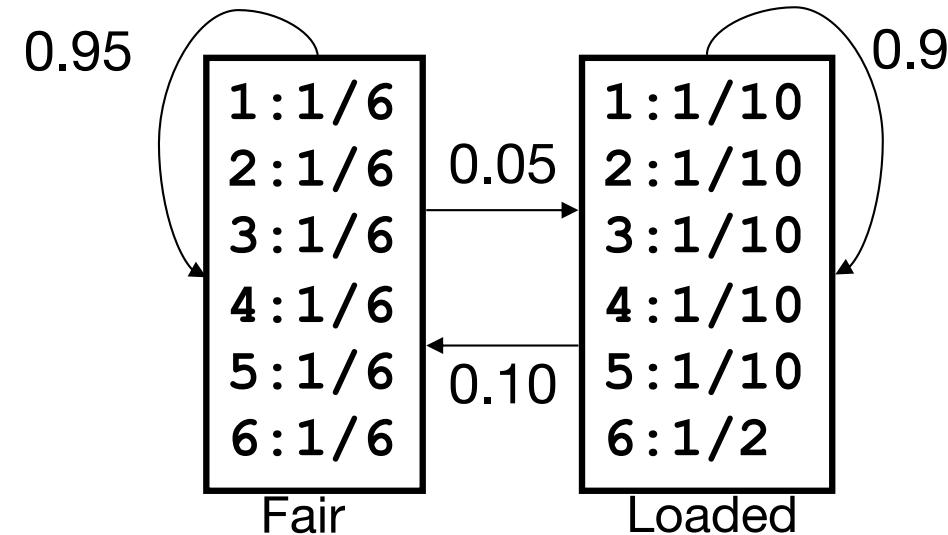


Deletion

Why hidden?

The unfair casino: Loaded die $p(6) = 0.5$; switch fair to load:0.05; switch load to fair: 0.1

- Model generates numbers
 - 312453666641
- Does not tell which die was used
- Alignment (decoding) can give the most probable solution/path (Viterbi)
 - FFFFFFFLLLLL
- Or most probable set of states
 - FFFFFFFLLLLL



HMM (a simple example)

ACA---**ATG**

TCAACT**ATC**

ACAC--**AGC**

AGA---**ATC**

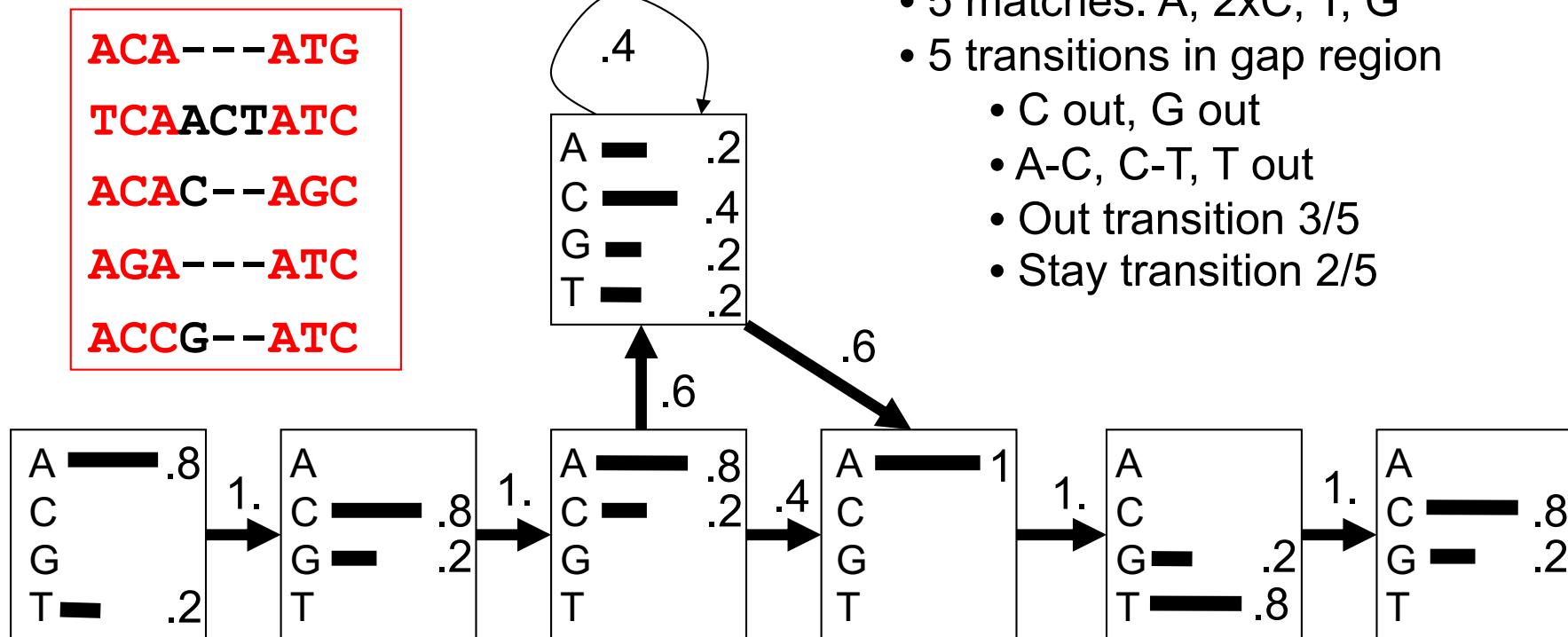
ACCG--**ATC**



Core of alignment

- Example from A. Krogh
- Core region defines the number of states in the HMM (**red**)
- Insertion and deletion statistics are derived from the non-core part of the alignment (black)

HMM construction (supervised learning)



$$\text{ACA---ATG} \quad 0.8 \times 1 \times 0.8 \times 1 \times 0.8 \times 0.4 \times 1 \times 1 \times 0.8 \times 1 \times 0.2 = 3.3 \times 10^{-2}$$

Scoring a sequence to an HMM

ACA---ATG $0.8 \times 1 \times 0.8 \times 1 \times 0.8 \times 0.4 \times 1 \times 0.8 \times 1 \times 0.2 = 3.3 \times 10^{-2}$

TCAACTATC $0.2 \times 1 \times 0.8 \times 1 \times 0.8 \times 0.6 \times 0.2 \times 0.4 \times 0.4 \times 0.4 \times 0.2 \times 0.6 \times 1 \times 1 \times 0.8 \times 1 \times 0.8 = 0.0075 \times 10^{-2}$

ACAC--AGC = 1.2×10^{-2}

Consensus:

ACAC--ATC = 4.7×10^{-2} , **ACA---ATC** = 13.1×10^{-2}

Exceptional:

TGCT--AGG = 0.0023×10^{-2}

Align sequence to HMM - Null model

- Score depends **strongly** on length
 - Null model is a random model. For length L the score is 0.25^L
 - Log-odds score for sequence S
 - $\text{Log}(\text{P}(S)/0.25^L)$
 - Positive score means more likely than Null model
- This is just like we did for PSSM $\text{log}(p/q)$!

ACA---ATG = 4.9

TCAACTATC = 3.0

ACAC--AGC = 5.3

AGA---ATC = 4.9

ACCG--ATC = 4.6

Consensus:

ACAC--ATC = 6.7

ACA---ATC = 6.3

Exceptional:

TGCT--AGG = -0.97

Note!

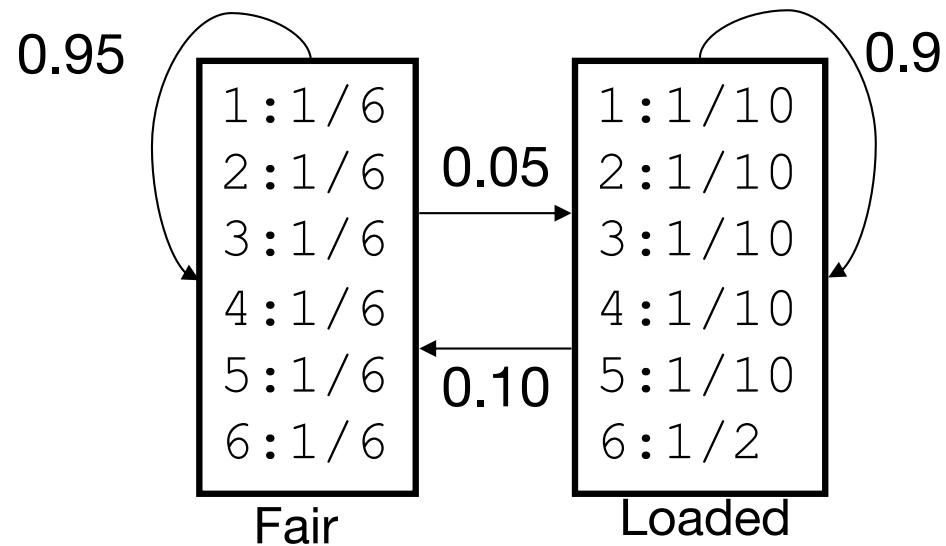
Aligning a sequence to an HMM

- Find the path through the HMM states that has the highest probability
 - For alignment, we found the path through the scoring matrix that had the highest sum of scores

The Viterbi algorithm

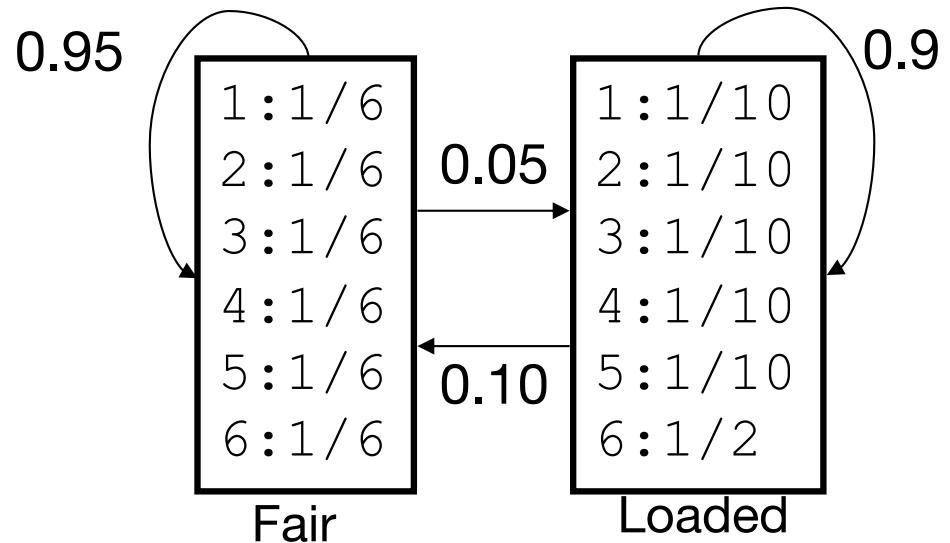
The unfair casino: Loaded dice $p(6) = 0.5$; switch fair to load:0.05; switch load to fair: 0.1. Start by randomly pick one of the two dice.

- Model generates numbers
 - 312453666641



Model decoding (Viterbi)

- Example: [566](#). What was the most likely series of dice used to generate this output?
- Use Brute force



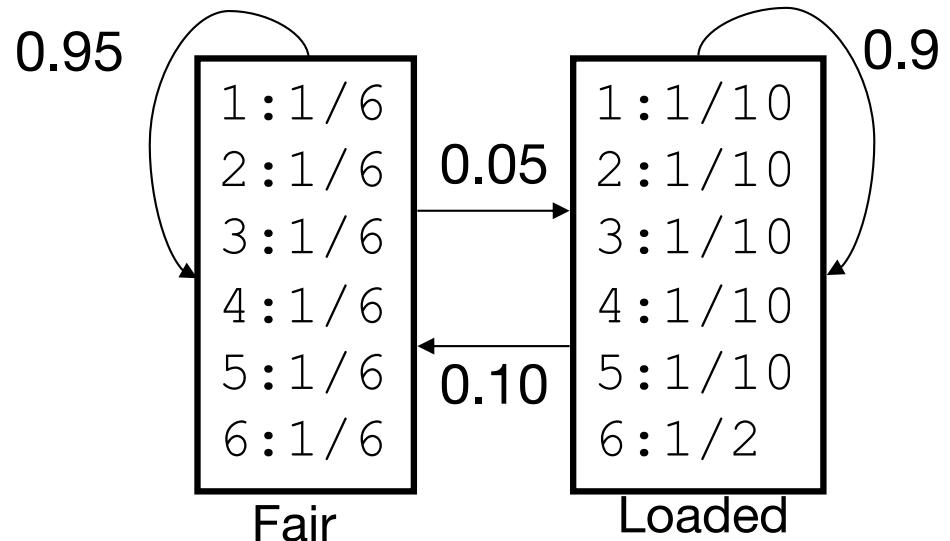
```
FFF = 0.5*0.167*0.95*0.167*0.95*0.167 = 0.0021
FFL = 0.5*0.167*0.95*0.167*0.05*0.5 = 0.00333
FLF = 0.5*0.167*0.05*0.5*0.1*0.167 = 0.000035
FLL = 0.5*0.167*0.05*0.5*0.9*0.5 = 0.00094
LFF = 0.5*0.1*0.1*0.167*0.95*0.167 = 0.00013
LFL = 0.5*0.1*0.1*0.167*0.05*0.5 = 0.000021
LLF = 0.5*0.1*0.9*0.5*0.1*0.167 = 0.00038
LLL = 0.5*0.1*0.9*0.5*0.9*0.5 = 0.0101
```

Aligning a sequence to an HMM

- Find the path through the HMM states that has the highest probability
 - For alignment, we found the path through the scoring matrix that had the highest sum of scores
- The number of possible paths rapidly gets very large making brute force search infeasible
 - Just like checking all path for alignment did not work
- Use dynamic programming
 - The Viterbi algorithm does the job

Model decoding (Viterbi)

- Example: [566](#). What was the most likely series of dice used to generate this output?
- Use Brute force



$$\text{FFF} = 0.5 * 0.167 * 0.95 * 0.167 * 0.95 * 0.167 = 0.0021$$

$$\text{FFL} = 0.5 * 0.167 * 0.95 * 0.167 * 0.05 * 0.5 = 0.00333$$

$$\text{FLF} = 0.5 * 0.167 * 0.05 * 0.5 * 0.1 * 0.167 = 0.000035$$

$$\text{FLL} = 0.5 * 0.167 * 0.05 * 0.5 * 0.9 * 0.5 = 0.00094$$

$$\text{LFF} = 0.5 * 0.1 * 0.1 * 0.167 * 0.95 * 0.167 = 0.00013$$

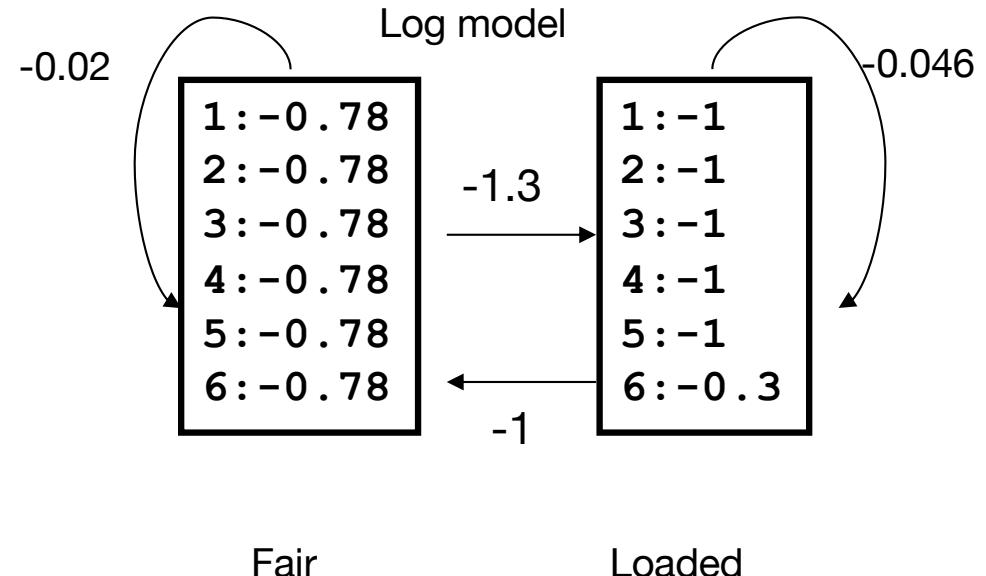
$$\text{LFL} = 0.5 * 0.1 * 0.1 * 0.167 * 0.05 * 0.5 = 0.000021$$

$$\text{LLF} = 0.5 * 0.1 * 0.9 * 0.5 * 0.1 * 0.167 = 0.00038$$

$$\text{LLL} = 0.5 * 0.1 * 0.9 * 0.5 * 0.9 * 0.5 = 0.0101$$

Or in log space

- Example: 566. What was the most likely series of dice used to generate this output?



$$\text{Log}(P(\text{LLL} | M)) = \log(0.5 * 0.1 * 0.9 * 0.5 * 0.9 * 0.5) = \log(0.0101)$$

or

$$\begin{aligned}\text{Log}(P(\text{LLL} | M)) &= \log(0.5) + \log(0.1) + \log(0.9) + \log(0.5) + \log(0.9) + \log(0.5) \\ &= -0.3 -1 -0.046 -0.3 -0.046 -0.3 = -1.99\end{aligned}$$

Model decoding (Viterbi)

- Example: 56611234. What was the most likely series of dice used to generate this output?

$$FF = 0.5 * 0.167 * 0.95 * 0.167$$

$$\text{Log}(FF) = -0.30 - 0.78 - 0.02 - 0.78 = \textcircled{-1.88}$$

$$LF = 0.5 * 0.1 * 0.1 * 0.167$$

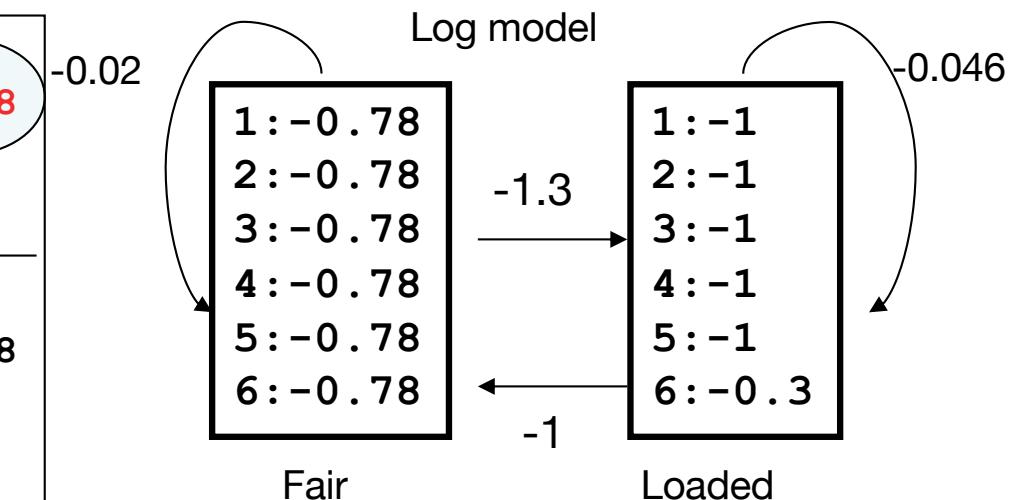
$$\text{Log}(LF) = -0.30 - 1 - 1 - 0.78 = -3.08$$

$$FL = 0.5 * 0.167 * 0.05 * 0.5$$

$$\text{Log}(FL) = -0.30 - 0.78 - 1.30 - 0.30 = -2.68$$

$$LL = 0.5 * 0.1 * 0.9 * 0.5$$

$$\text{Log}(LL) = -0.30 - 1 - 0.046 - 0.3 = \textcircled{-1.65}$$

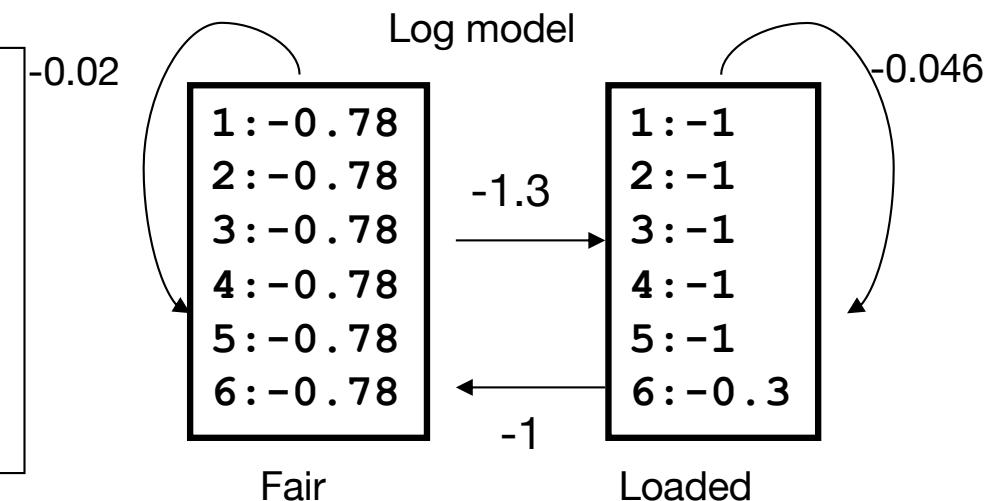


	5	6	6	6	1	1	2	3	4
F	-1.08	-1.88							
L	-1.30	-1.65							

Model decoding (Viterbi)

- Example: 566611234. What was the most likely series of dice used to generate this output?

$\text{FFF} = 0.5 * 0.167 * 0.95 * 0.167 * 0.95 * 0.167 = 0.0021$
 $\text{FLF} = 0.5 * 0.167 * 0.05 * 0.5 * 0.1 * 0.167 = 0.000035$
 $\text{LFF} = 0.5 * 0.1 * 0.1 * 0.167 * 0.95 * 0.167 = 0.00013$
 $\text{LLF} = 0.5 * 0.1 * 0.9 * 0.5 * 0.1 * 0.167 = 0.00038$
 $\text{FLL} = 0.5 * 0.167 * 0.05 * 0.5 * 0.9 * 0.5 = 0.00094$
 $\text{FFL} = 0.5 * 0.167 * 0.95 * 0.167 * 0.05 * 0.5 = 0.00333$
 $\text{LFL} = 0.5 * 0.1 * 0.1 * 0.167 * 0.05 * 0.5 = 0.000021$
 $\text{LLL} = 0.5 * 0.1 * 0.9 * 0.5 * 0.9 * 0.5 = 0.0101$

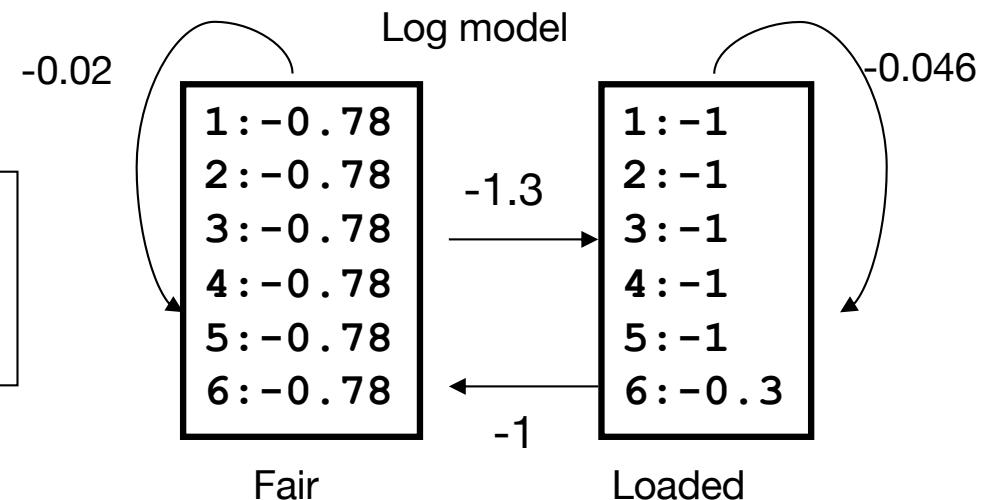


	5	6	6	6	1	1	2	3	4
F	-1.08	-1.88							
L	-1.30	-1.65							

Model decoding (Viterbi)

- Example: 566611234. What was the most likely series of dice used to generate this output?

$$\begin{aligned} \text{FFF} &= 0.5 * 0.167 * 0.95 * 0.167 * 0.95 * 0.167 = 0.0021 \\ \text{Log}(P(\text{FFF})) &= -2.68 \\ \text{LLL} &= 0.5 * 0.1 * 0.9 * 0.5 * 0.9 * 0.5 = 0.0101 \\ \text{Log}(P(\text{LLL})) &= -1.99 \end{aligned}$$

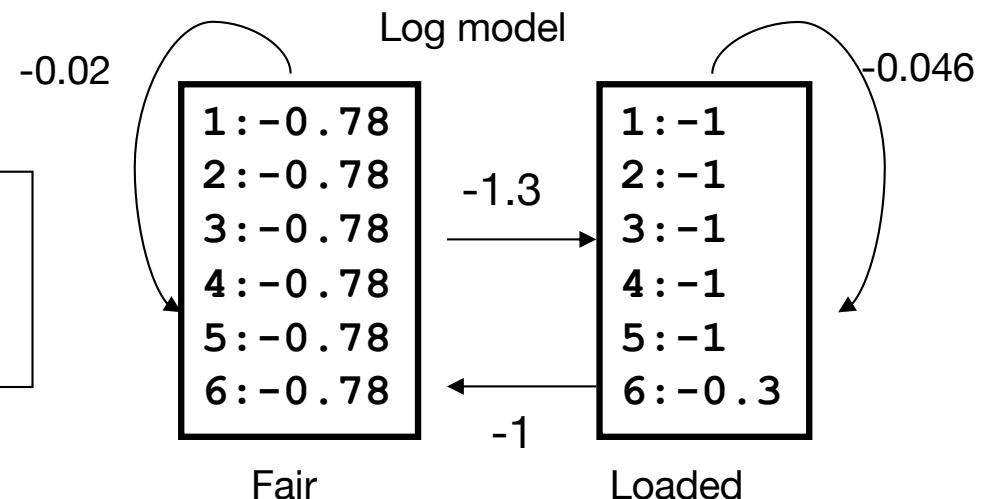


	5	6	6	6	1	1	2	3	4
F	-1.08	-1.88	-2.68						
L	-1.30	-1.65	-1.99						

Model decoding (Viterbi)

- Example: 566611234. What was the most likely series of dice used to generate this output?

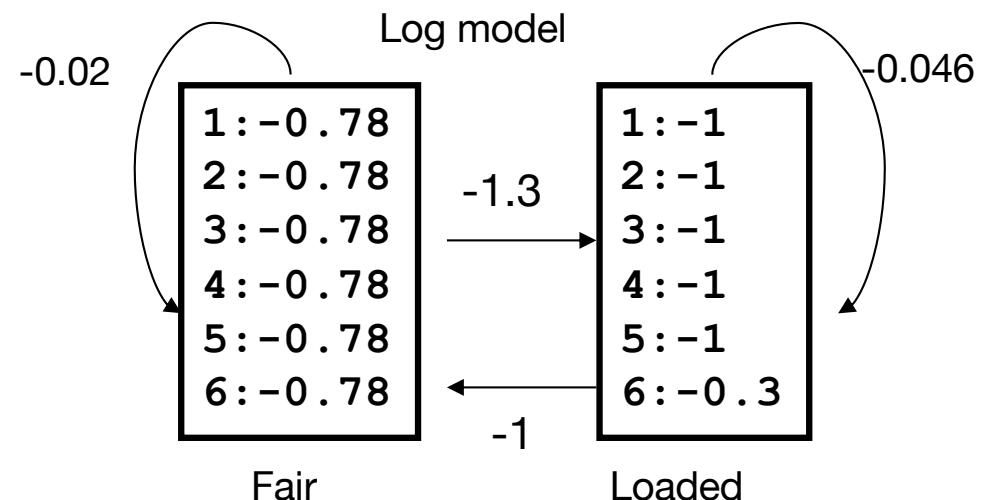
$$\begin{aligned}
 F_{i+1}(6) &= F_i + F \rightarrow F + P_F(6) \text{ or} \\
 L_i &+ L \rightarrow F + P_F(6) \\
 F_{i+1}(6) &= -1.88 + -0.02 + -0.78 = -2.68 \text{ or} \\
 -1.65 &+ -1 + -0.78 = -3.43
 \end{aligned}$$



	i	i+1	Fair	Loaded
	5	6	6	6
F	-1.08	-1.88	-2.68	
L	-1.30	-1.65	-1.99	

Model decoding (Viterbi)

- Example: 566611234. What was the most likely series of dice used to generate this output?



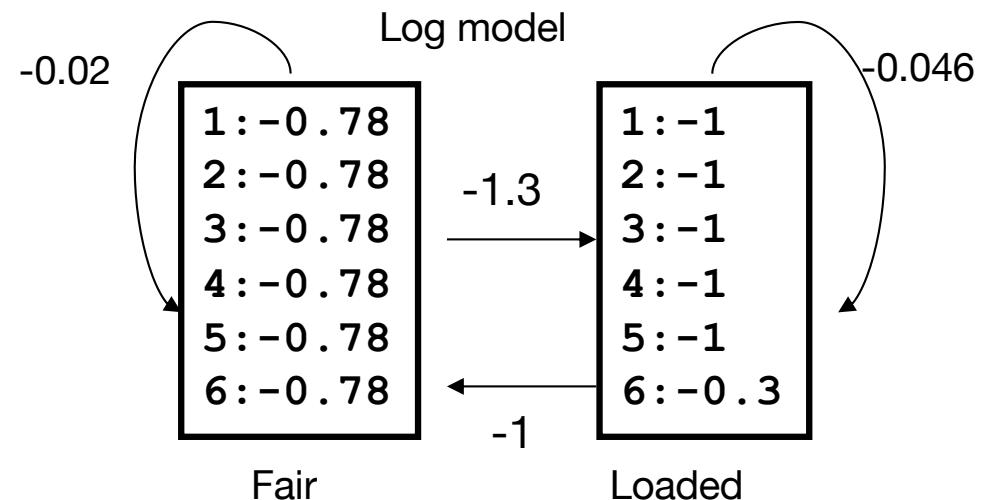
	5	6	6	6	1	1	2	3	4
F	-1.08	-1.88	-2.68						
L	-1.30	-1.65	-1.99						

Model decoding (Viterbi)

- Example: 566611234. What was the most likely series of dice used to generate this output?

$$-0.78 - 0.02 - 2.68 = -3.48$$

$$-0.78 - 1 - 1.99 = -3.77$$



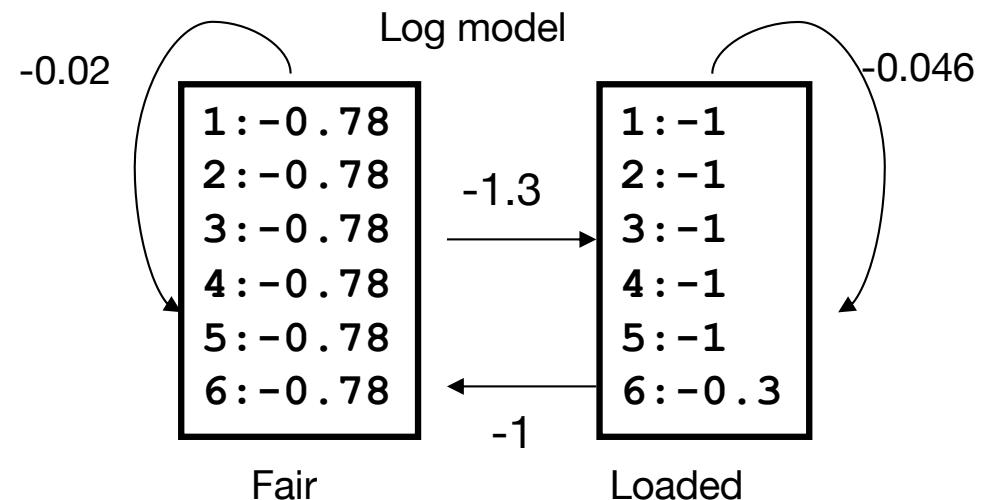
	5	6	6	6	1	1	2	3	4
F	-1.08	-1.88	-2.68	-					
L	-1.30	-1.65	-1.99						

Model decoding (Viterbi)

- Example: 566611234. What was the most likely series of dice used to generate this output?

$$-0.78 - 0.02 - 2.68 = -3.48$$

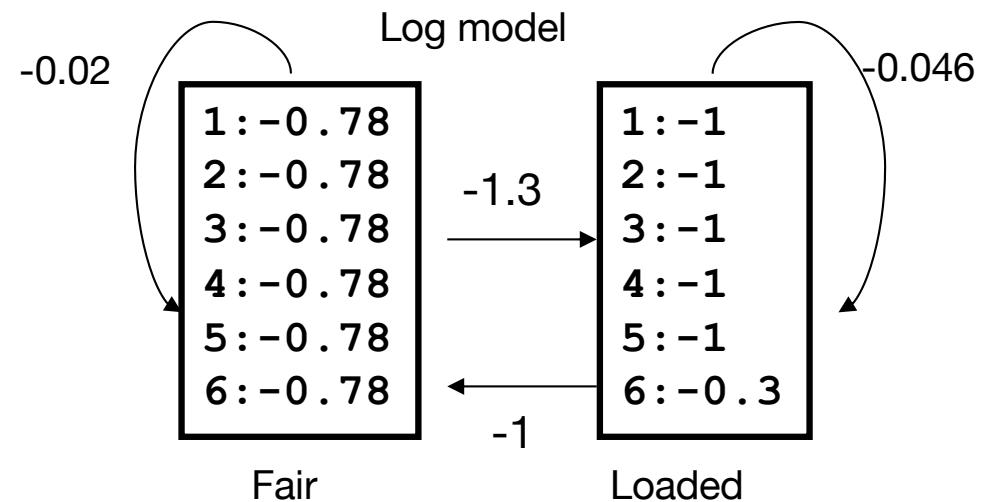
$$-0.78 - 1 - 1.99 = -3.77$$



	5	6	6	6	1	1	2	3	4
F	-1.08	-1.88	-2.68	-3.48					
L	-1.30	-1.65	-1.99						

Model decoding (Viterbi)

- Now we can formalize the algorithm!



$$P_l(i+1) = p_l(i+1) \cdot \max_k (P_k(i) \cdot a_{kl}) \quad or$$

$$\log(P_l(i+1)) = \log(p_l(i+1)) + \max_k (\log(P_k(i)) + \log(a_{kl}))$$

New match

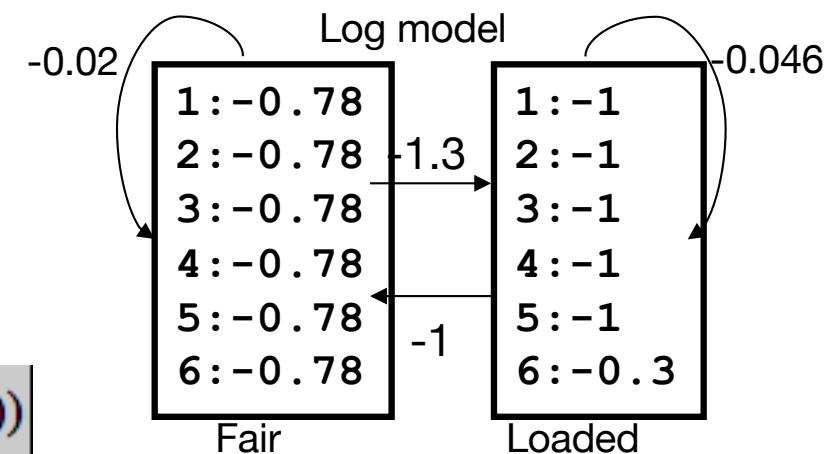
Old max score

Transition

Model decoding (Viterbi). Can you do it?

- Example: 566611234. What was the most likely series of dice used to generate this output?
- Fill out the table using the Viterbi recursive algorithm
 - Add the arrows for backtracking
- Find the optimal path

$$\log(P_i(i+1)) = \log(p_i(i+1)) + \max_k(\log(P_k(i)) + \log(a_{kl}))$$

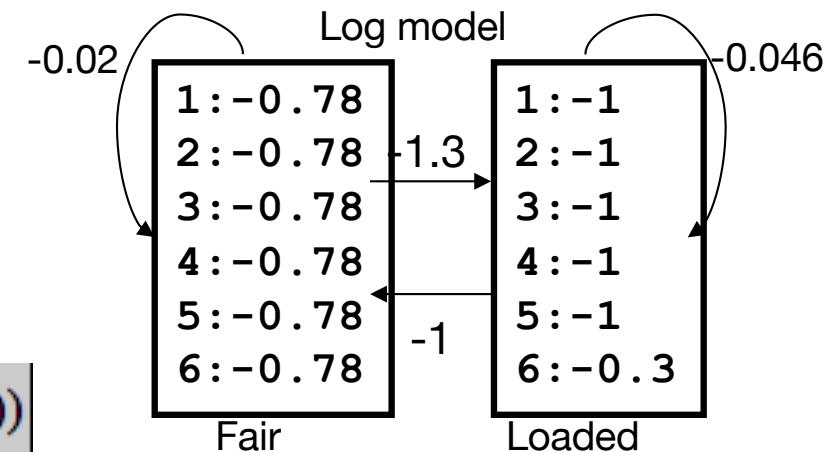


	5	6	6	6	1	1	2	3	4
F	-1.08	-1.88	-2.68	-3.48					
L	-1.30	-1.65	-1.99						

Model decoding (Viterbi). Can you do it?

- Example: 566611234. What was the most likely series of dice used to generate this output?
- Fill out the table using the Viterbi recursive algorithm
 - Add the arrows for backtracking
- Find the optimal path

$$\log(P_i(i+1)) = \log(p_i(i+1)) + \max_k (\log(P_k(i)) + \log(a_{kl}))$$



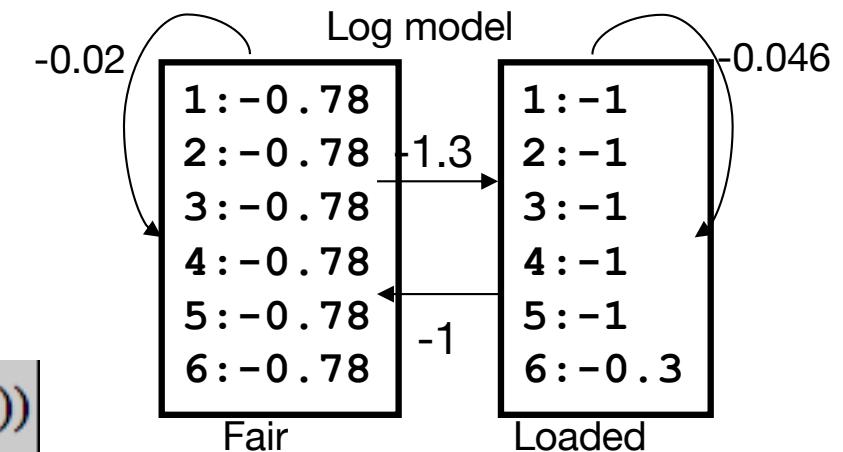
$$-0.78 + \max(\underline{-2.68 - 0.02}, \underline{-1.99 - 1})$$

	5	6	6	6	1	1	2	3	4
F	-1.08	-1.88	<u>-2.68</u>	<u>-3.48</u>					
L	-1.30	-1.65	-1.99						

Model decoding (Viterbi). Can you do it?

- Example: 566611234. What was the most likely series of dice used to generate this output?
- Fill out the table using the Viterbi recursive algorithm
 - Add the arrows for backtracking
- Find the optimal path

$$\log(P_l(i+1)) = \log(p_l(i+1)) + \max_k (\log(P_k(i)) + \log(a_{kl}))$$

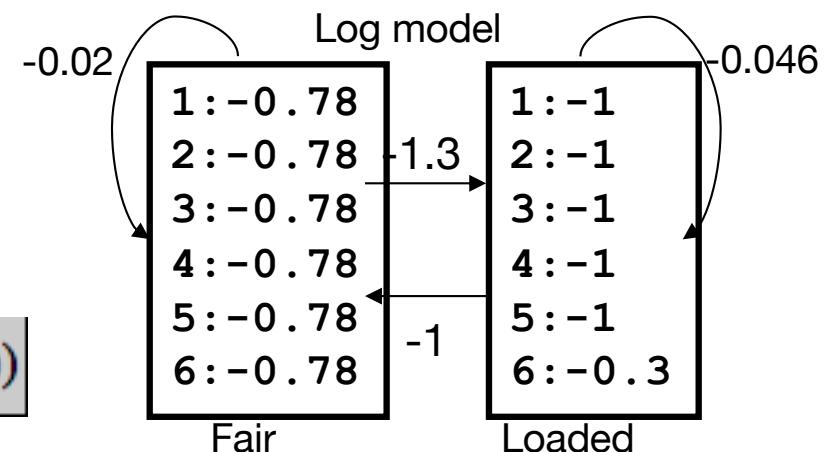


	5	6	6	6	1	1	2	3	4
F	-1.08	-1.88	-2.68	-3.48		-4.92		-6.53	
L	-1.30	-1.65	-1.99		-3.39			-6.52	

Model decoding (Viterbi). Can you do it?

- Example: 566611234. What was the most likely series of dice used to generate this output?
- Fill out the table using the Viterbi recursive algorithm
 - Add the arrows for backtracking
- Find the optimal path

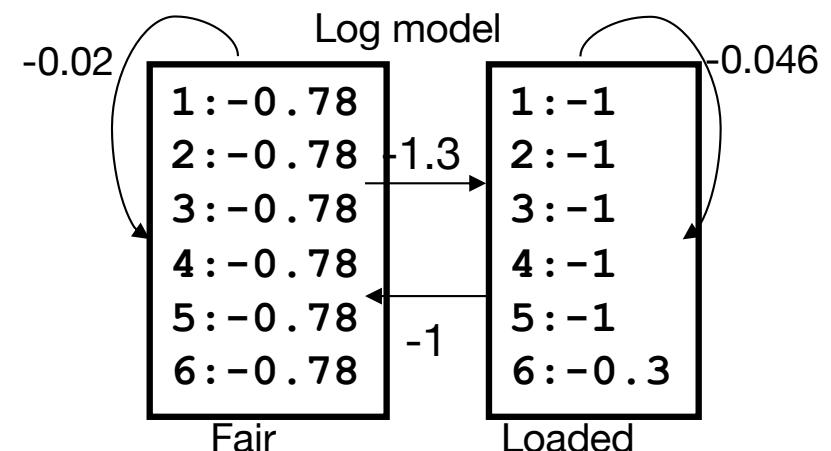
$$\log(P_l(i+1)) = \log(p_l(i+1)) + \max_k (\log(P_k(i)) + \log(a_{kl}))$$



	5	6	6	6	1	1	2	3	4
F	-1.08	-1.88	-2.68	-3.48	-4.12	-4.92	5.73	-6.53	-7.33
L	-1.30	-1.65	-1.99	-2.34	-3.39	-4.44	-5.48	-6.52	-7.57

Model decoding (Viterbi). Can you do it?

- Example: 566611234. What was the most likely series of dice used to generate this output?
- The most likely path is
 - LLLLFFFFFF



	5	6	6	6	1	1	2	3	4
F	-1.08	-1.88	-2.68	-3.48	-4.12	-4.92	5.73	-6.53	-7.33
L	-1.30	-1.65	-1.99	-2.34	-3.39	-4.32	-5.48	-6.52	-7.57

Model decoding (Viterbi).

- What happens if you have three dice?

	5	6	6	6	1	1	2	3	4
F	-1.0								
L1	-1.2								
L2	-1.3								

$$\log(P_l(i+1)) = \log(p_l(i+1)) + \max_k(\log(P_k(i)) + \log(a_{kl}))$$

And if you have a trans-membrane model

- What is the most likely path (alignment) of a protein sequence to the model

	D	G	V	L	I	M	A	D	Q
iC	-1.0								
M	-1.2								
xC	-1.3								

$$\log(P_l(i+1)) = \log(p_l(i+1)) + \max_k(\log(P_k(i)) + \log(a_{kl}))$$

HMM's and weight matrices

- In the case of un-gapped alignments HMM's become simple weight matrices
- To achieve high performance, the emission frequencies are estimated using the techniques of
 - Sequence weighting
 - Pseudo counts

Profile HMM's

- Alignments based on conventional scoring matrices (BLOSUM62) scores all positions in a sequence in an equal manner
- Some positions are highly conserved, some are highly variable (more than what is described in the BLOSUM matrix)
- Profile HMM's are ideal suited to describe such position specific variations

Sequence profiles

Conserved deletion Non-conserved Insertion

The diagram shows two protein sequences aligned vertically. Regions of the top sequence are highlighted with light blue boxes. Red text annotations above the sequence identify these as 'Conserved' (covering the first few residues), 'deletion' (covering a gap where a residue is missing), and 'Non-conserved' (covering a region with a different amino acid) 'Insertion' (covering a region where an extra residue is present). The bottom sequence is also shown with a light blue box highlighting a region. The sequences themselves are composed of standard amino acid abbreviations.

ADDGSLAFVPSEF--SISPG**E**KIVFKNNAGFPNIVFD**E**DSIPSGVDASKI**S**M**S****E****D****L****I****N**
TVNGAI--PGPLIAERL**K**E**G**QNVRVTNT**L**DEDTSIHWHGLLVPGMDGVPGVSFPG---I
-TSMAPAFGVQEFYRTVK**Q**G**D**EV**T**VTIT----NIDQIED-VSHGFVVVNHGVSME---I
IE--KMKYL**T**PEVFYT**I**KAGETVYWVNGEVMPHNVA**F**KKGIV--GEDAFRG**E**MM**T****D**---
-TSVAPSFSQPSF-LTV**K**E**G**DEV**T**V**I**VTNLDE-----IDDLTHGFTMGNHG**V**AME---V
ASAETMV**F**EPDFLV**I**GPGRVRFVPTHK-SHNAAT**I**DGMVPEGVEGF**K**SRINDE---
TVNGQ--FPGPRIAGVAREGD**Q**VLV**K**VVN**H**VA**E**N**I**TI**H**WHGVQLGTGWADGPAY**V**T**Q****C****P**
TKAVVLT**F**NTSVEICLVM**Q****G**T**S****I**V---AA**E**SHPLHLHG**F**PSNFNL**V**D**P****M****E**RNTAGVP

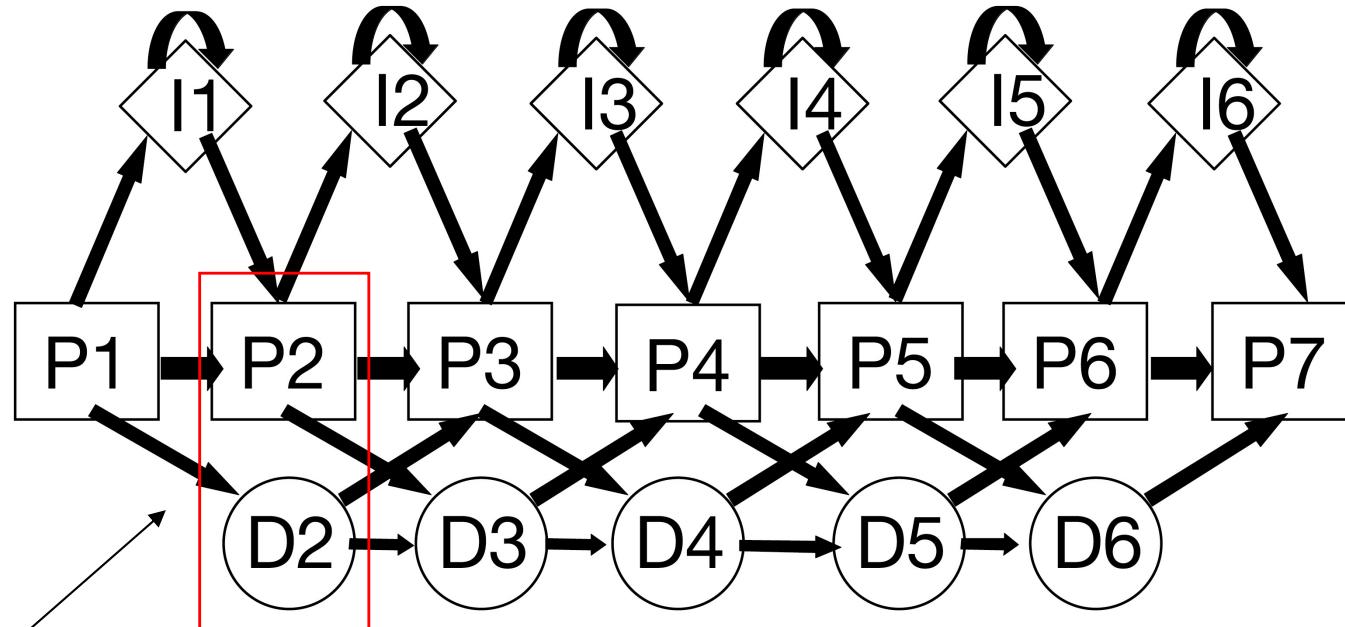
Matching any thing
but G => large
negative score

Any thing can match

HMM vs. alignment

- Detailed description of core
 - Conserved/variable positions
- Price for insertions/deletions varies at different locations in sequence
- These features cannot be captured in conventional alignments

Profile HMM's



All P/D pairs must
be visited once

$$\begin{aligned} & L_1 - Y_2 A_3 V_4 R_5 - I_6 \\ & P_1 D_2 P_3 P_4 I_4 P_5 D_6 P_7 \end{aligned}$$

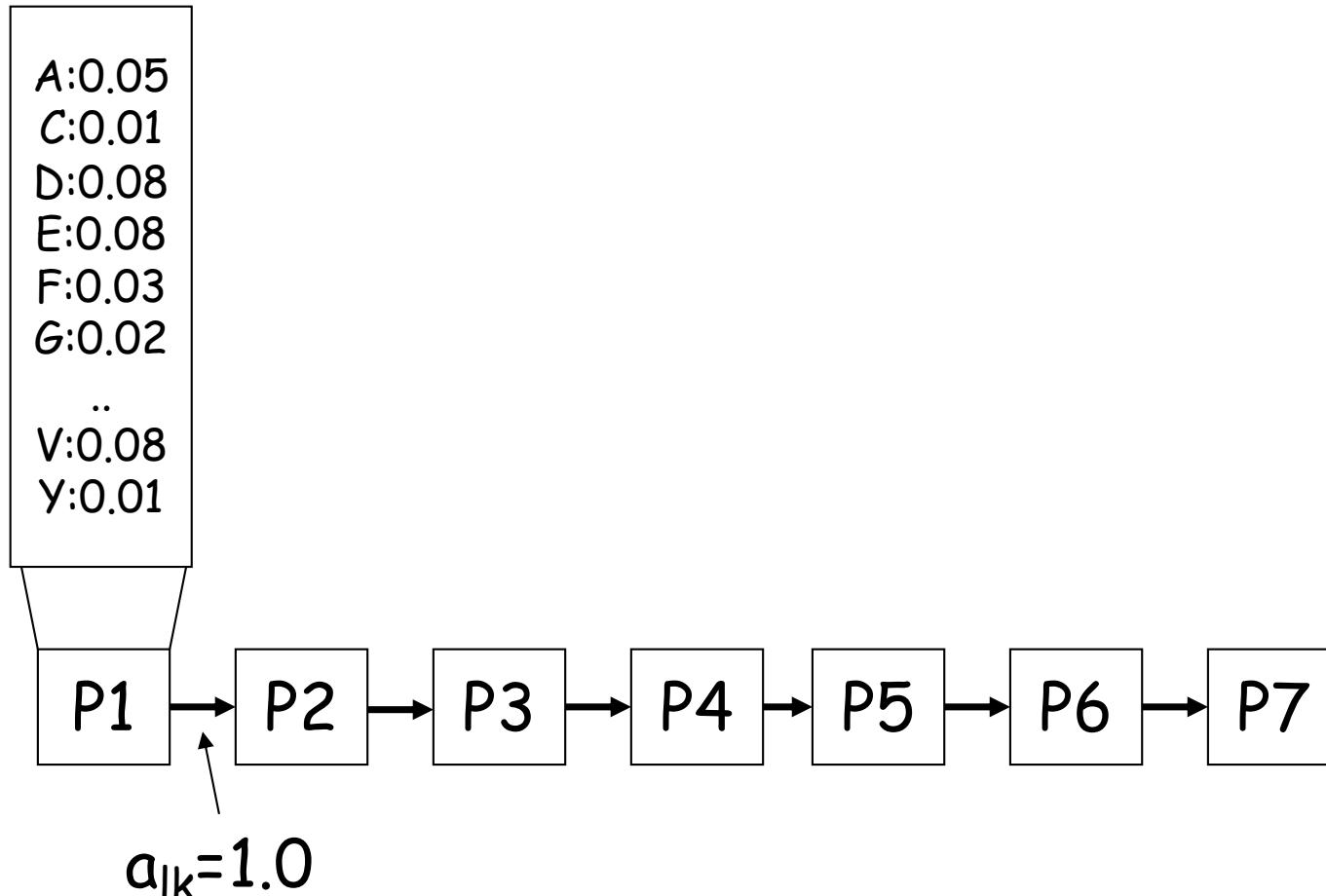
Profile HMM

- Un-gapped profile HMM is just a sequence profile



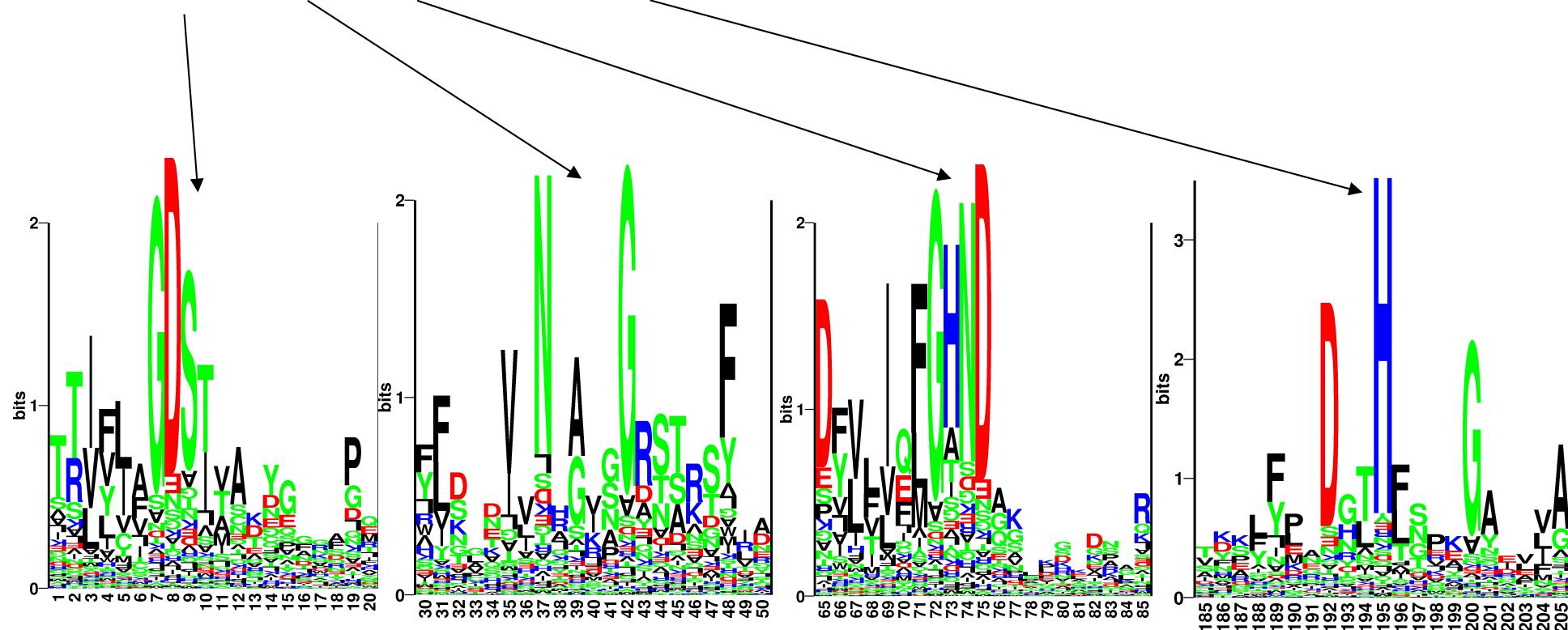
Profile HMM

- Un-gapped profile HMM is just a sequence profile



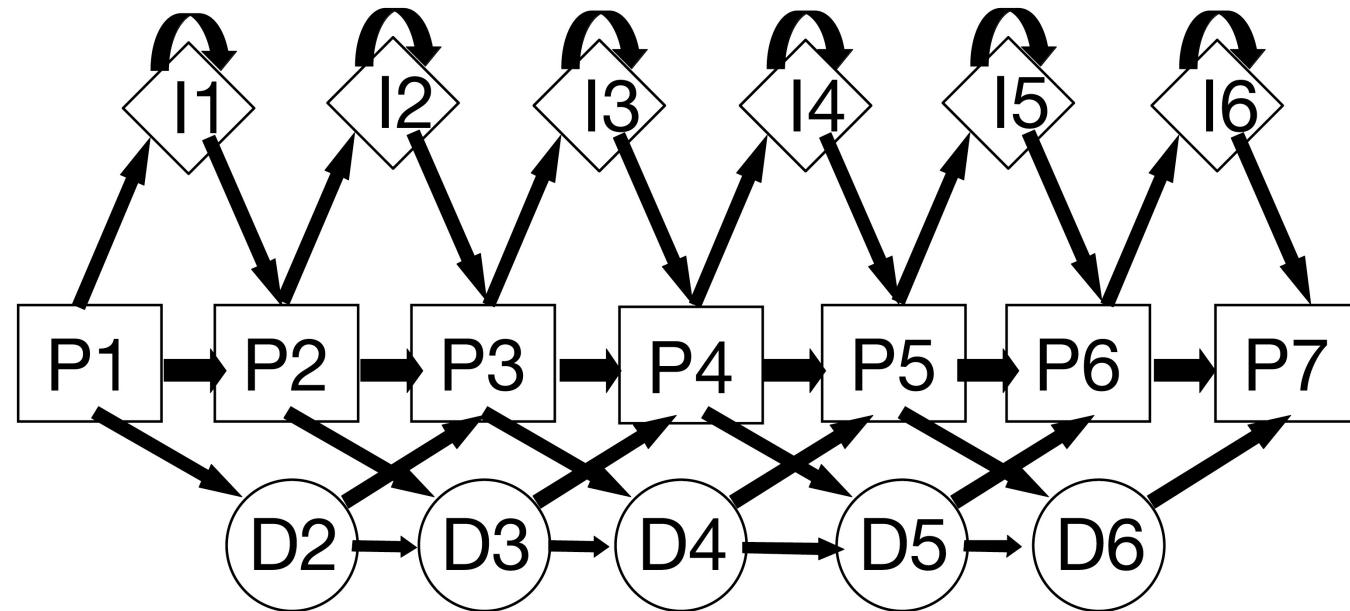
Example. Where is the active site?

- Sequence profiles might show you where to look!
- The active site could be around
 - S9, G42, N74, and H195



Profile HMM

- Profile HMM (deletions and insertions)



Profile HMM (deletions and insertions)

QUERY	HAMDIRCYHSGG	PLHL	GEI	EDF	NGQSCIVCPWHKYKITLATGE	GLY	QSINPKDPS
Q8K2P6	HAMDIRCYHSGG	PLHL	GEI	EDF	NGQSCIVCPWHKYKITLATGE	GLY	QSINPKDPS
Q8TAC1	HAMDIRCYHSGG	PLHL	GDI	EDF	DGRPCIVCPWHKYKITLATGE	GLY	QSINPKDPS
Q07947	FAVQDTCTHGDW	ALSE	GYI	DGD	-	VVECTLHFGKFCVRTGK	VKAL-----PA
P0A185	YATDNLCTHGSA	RMSD	GYI	EGRE	-	IECPLHQGRFDVCTGK	ALC-----APV
P0A186	YATDNLCTHGSA	RMSD	GYI	EGRE	-	IECPLHQGRFDVCTGK	ALC-----APV
Q51493	YATDNLCTHGAA	RMSD	GFI	EGRE	-	IECPLHQGRFDVCTGR	ALC-----APV
A5W4F0	FAVQDTCTHGDW	ALSD	GYI	DGD	-	IVECTLHFGKFCVRTGK	VKAL-----PA
P0C620	FAVQDTCTHGDW	ALSD	GYI	DGD	-	IVECTLHFGKFCVRTGK	VKAL-----PA
P08086	FAVQDTCTHGDW	ALSD	GYI	DGD	-	IVECTLHFGKFCVRTGK	VKAL-----PA
Q52440	FATQDQCTHGEW	SLSE	GGY	LDGD	-	VVECSLHMGKFCVRTGK	-V
Q7N4V8	FAVDDRCSHGNA	SISE	GYI	ED	-	NATVECPLHTASFCLRTGK	ALCL-----PA
P37332	FATQDRCTHGDW	SLSDG	GYI	EGD	-	VVECSLHMGKFCVRTGK	-V
A7ZPY3	YAINDRCSHGNA	SMSE	GYI	EDD	-	ATVECPLHAASFCLKTGK	ALCL-----PA
P0ABW1	YAINDRCSHGNA	SMSE	GYI	EDD	-	ATVECPLHAASFCLKTGK	ALCL-----PA
A8A346	YAINDRCSHGNA	SMSE	GYI	EDD	-	ATVECPLHAASFCLKTGK	ALCL-----PA
P0ABW0	YAINDRCSHGNA	SMSE	GYI	EDD	-	ATVECPLHAASFCLKTGK	ALCL-----PA
P0ABW2	YAINDRCSHGNA	SMSE	GYI	EDD	-	ATVECPLHAASFCLKTGK	ALCL-----PA
Q3YZ13	YAINDRCSHGNA	SMSE	GYI	EDD	-	ATVECPLHAASFCLKTGK	ALCL-----PA
Q06458	YALDNLEPGSEANVLSR	GLI	-	GDAGGEPIVISPLYKQRIRLRDG	-	-	-



Core

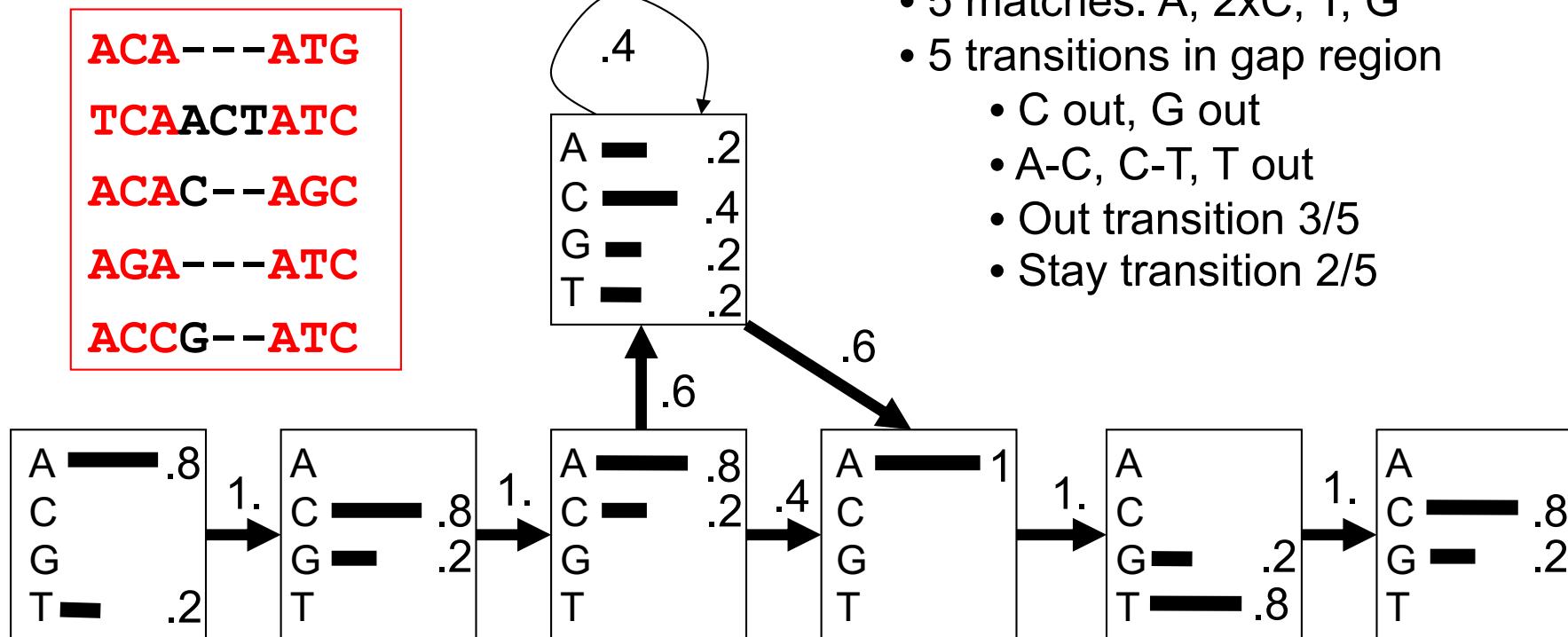


Insertion



Deletion

HMM construction (supervised learning)



$$\text{ACA---ATG} \quad 0.8 \times 1 \times 0.8 \times 1 \times 0.8 \times 0.4 \times 1 \times 1 \times 0.8 \times 1 \times 0.2 = 3.3 \times 10^{-2}$$

The HMMer program

- HMMer is a open source program suite for profile HMM for biological sequence analysis
- Used to make the Pfam database of protein families
 - <http://pfam.sanger.ac.uk/>

A HMMer example

- Example from the CASP8 competition
- What is the best PDB template for building a model for the sequence T0391

```
>T0391 rieske ferredoxin, mouse, 157 residues
SDPEISEQDEEKKYTSVCVGREEDIRKSERMTAVVHDREVVI FYHKGEYHAMDIRCYHS
GGPLHLGEIEDFNGQSCIVCPWHKYKITLATGEGLYQSINPKDPSAKPKWCSKGVKQRIH
TVKVDNGNIYVTLSKEPFKCDSDYYATGEFKVIQSSS
```

A HMMer example

- What is the best PDB template for building a model for the sequence T0391
- Use Blast
 - No hits
- Use Psi-Blast
 - No hits
- Use Hmmer

A HMMer example

- Use Hmmer
 - Make multiple alignment using Blast
 - Make model using
 - hmmbuild
 - hmmcalibrate
 - Find PDB template using
 - hmmsearch

A HMMer example

- Make multiple alignment using Blast

```
blastpgp -j 3 -e 0.001 -m 6 -i T0391.fsa -d sp -b  
10000000 -v 10000000 > T0391.fsa.blastout
```

- Make Stockholm format

```
# STOCKHOLM 1.0  
QUERY DPEISEQDEEKKYTSVCVGREEDIRKS-ERMTAVVHD-RE--V-V-IF--Y-H-KGE-Y  
Q8K2P6 DPEISEQDEEKKYTSVCVGREEDIRKS-ERMTAVVHD-RE--V-V-IF--Y-H-KGE-Y  
Q8TAC1 ----SAQDPEKREYSSVCVGREDDIKKS-ERMTAVVHD-RE--V-V-IF--Y-H-KGE-Y
```

- Build HMM

```
hmmbuild T0391.hmm T0391.fsa.blastout.sto
```

- (Calibrate HMM (to estimate correct p-values))
hmmpcalibrate T0391.hmm) In older version of HMMER

- Search for templates in PDB

```
hmmpsearch T0391.hmm pdb > T0391.out
```

A HMMer example

Sequence	Description	Score	E-value	N
2E4Q.A	mol:aa ELECTRON TRANSPORT	163.7	6.7e-45	1
2E4P.B	mol:aa ELECTRON TRANSPORT	163.7	6.7e-45	1
2E4P.A	mol:aa ELECTRON TRANSPORT	163.7	6.7e-45	1
2E4Q.C	mol:aa ELECTRON TRANSPORT	163.7	6.7e-45	1
2YVJ.B	mol:aa OXIDOREDUCTASE/ELECTRON TRANSPORT	163.7	6.7e-45	1
1FQT.A	mol:aa OXIDOREDUCTASE	160.9	4.5e-44	1
1FQT.B	mol:aa OXIDOREDUCTASE	160.9	4.5e-44	1
2QPZ.A	mol:aa METAL BINDING PROTEIN	137.3	5.6e-37	1
2Q3W.A	mol:aa ELECTRON TRANSPORT	116.2	1.3e-30	1
1VM9.A	mol:aa ELECTRON TRANSPORT	116.2	1.3e-30	1

Validation. CE structural alignment

CE 2E4Q A 3D89 A (run on IRIX machines at CBS)

Structure Alignment Calculator, version 1.01, last modified: May 25, 2000.

CE Algorithm, version 1.00, 1998.

Chain 1: /usr/cbs/bio/src/ce_distr/data.cbs/pdb2e4q.ent:A (Size=109)

Chain 2: /usr/cbs/bio/src/ce_distr/data.cbs/pdb3d89.ent:A (Size=157)

Alignment length = 101 Rmsd = 2.03A Z-Score = 5.5 Gaps = 20 (19.8%)

CPU = 1s Sequence identities = 18.1%

Chain 1: 2 TFTKACSVDEVPPGEALQVSHDAQKVAIFNVDGEFFATQDQCTHGEWSLSEGGYLDG----DVVECSLHM

Chain 2: 16 TSVCVGREEDIRKSERMTAVVHDREVVIFYHKGEYHAMDIRCYHSGGPLH-LGEIEDFNGQSCIVCPWHK

Chain 1: 68 GKFCVRTGKVKS-----PPPC-----EPLKVYPIRIEGRDVLVDFSRAALH

Chain 2: 85 YKITLATGEGLYQSINPKDPSAKPKWCSKGVKQRIHTVKVDNGNIYVTL-SKEPF

HMM packages

- HMMER (<http://hmmer.wustl.edu/>)
 - S.R. Eddy, WashU St. Louis. Freely available.
 - SAM (<http://www.cse.ucsc.edu/research/compbio/sam.html>)
 - R. Hughey, K. Karplus, A. Krogh, D. Haussler and others, UC Santa Cruz. Freely available to academia, nominal license fee for commercial users.
 - META-MEME (<http://metameme.sdsc.edu/>)
 - William Noble Grundy, UC San Diego. Freely available. Combines features of PSSM search and profile HMM search.
 - NET-ID, HMMpro (<http://www.netid.com/html/hmmpro.html>)
 - Freely available to academia, nominal license fee for commercial users.
 - Allows HMM architecture construction.
 - EasyGibbs (<http://www.cbs.dtu.dk/biotools/EasyGibbs/>)
 - Webserver for Gibbs sampling of proteins sequences
-