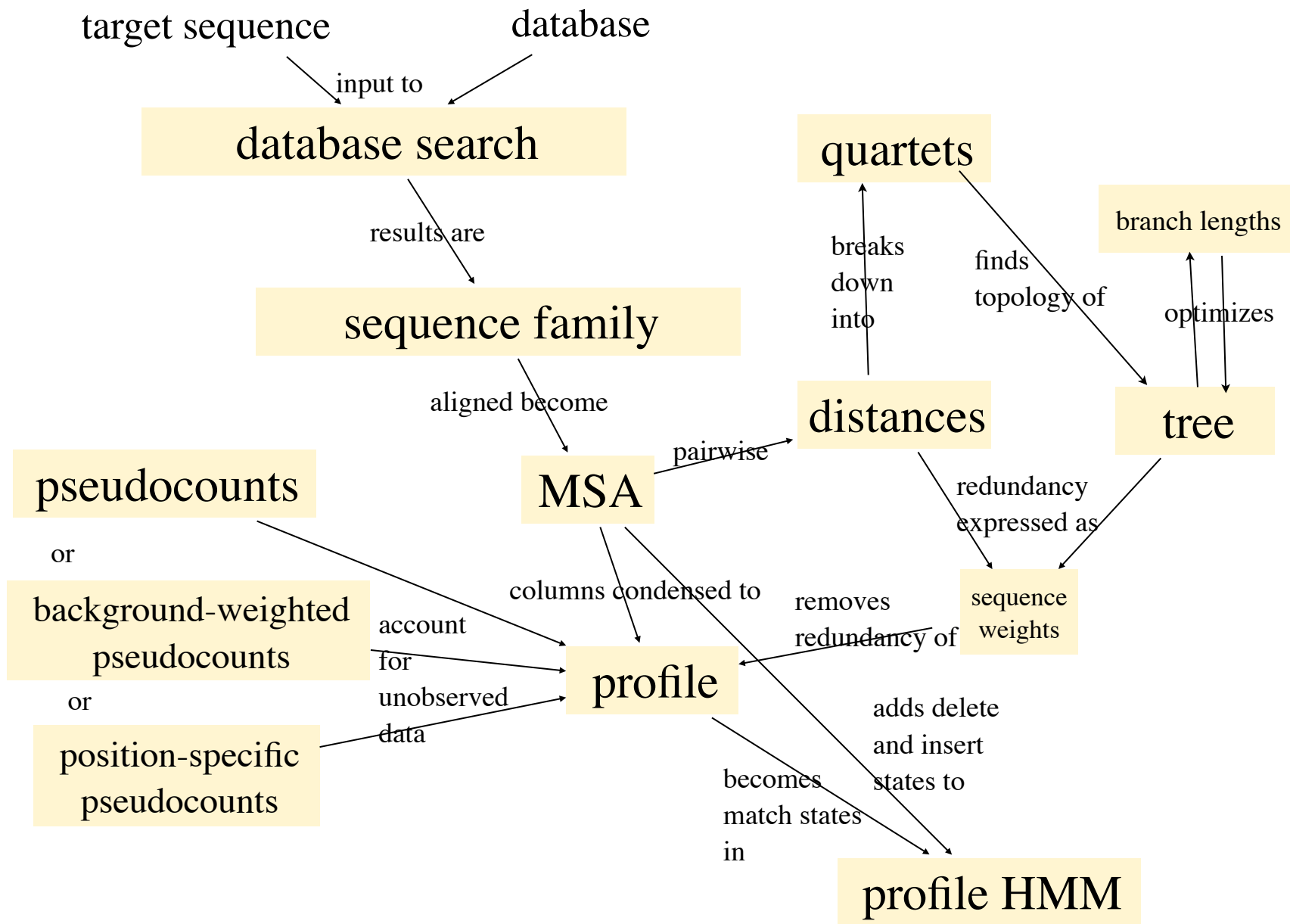


# Bioinformatics 1--lecture 17

Markov chains

Hidden Markov models

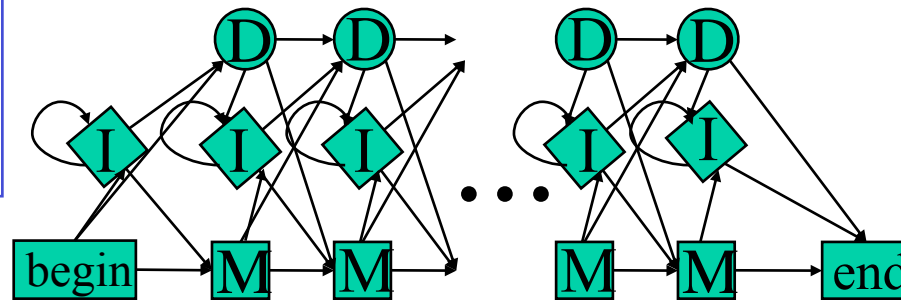
Profile HMMs



**overview**

# Profile hidden Markov models

I = insertion state  
D = deletion state  
M = match state



The probability of a **gap** or **insertion** might be position specific.  
Profile HMMs can model this.

# Markov processes

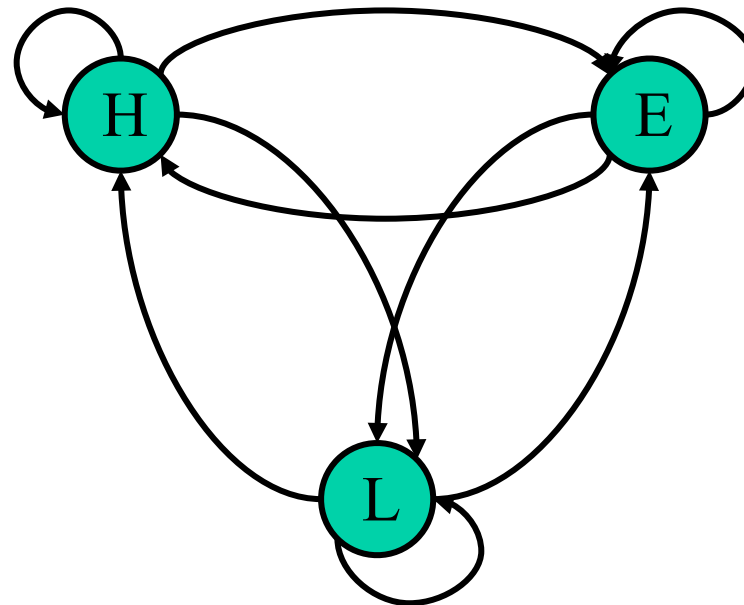


Markov process is any process where the next item in the list depends on the current item. The dimension can be time, sequence position, etc

# Modeling proteins using Markov chains

A Markov chain is a network of “states” connected by “transitions”

A Markov chain is a stochastic model that “emits” symbol data whose probability depends only on the last symbol emitted.



H=helix

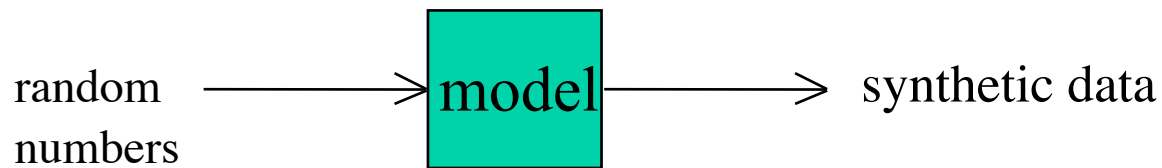
E=extended (strand)

L=loop

# What is a stochastic model?

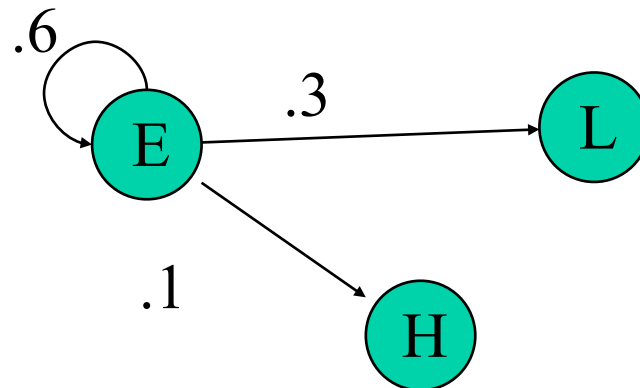
A model is a simplified version of reality. The simpler, the better.

*A stochastic model* has the form:



# Markov states

- ...*emits* a symbol each time you visit it.
- ...*connects* to other states (and possibly itself), with probabilities attached.



The sum of all transition probabilities = 1

note ==> Markov chains emit **discrete 1-dimensional** data.



# Bayes' notation and Rabiner's notation

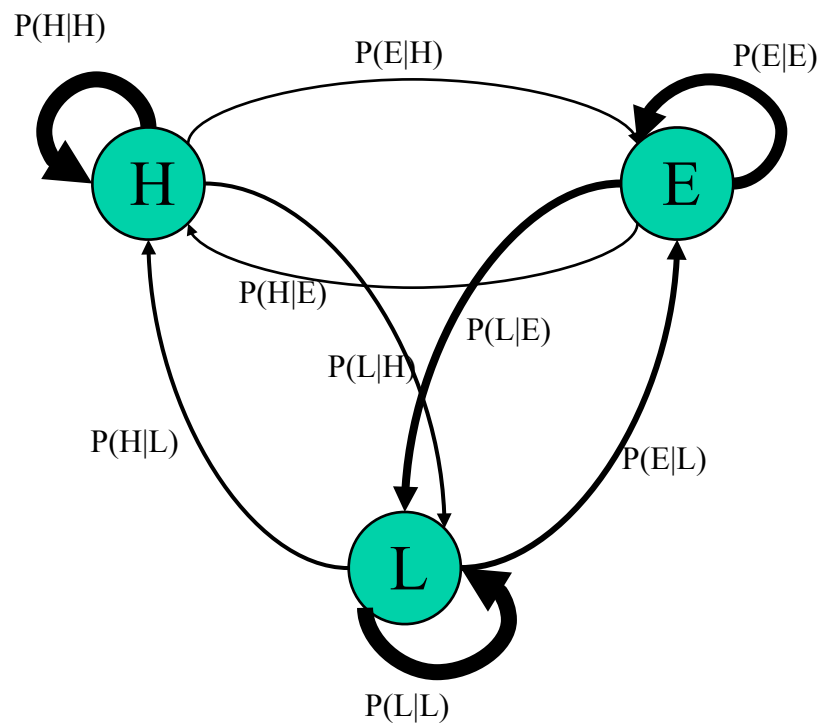
$$a_{yx} = P(x | y) = \frac{P(y, x)}{P(y)} = \frac{F(y, x)}{F(y)}$$

...the conditional probability of  $x$  given  $y$ .

$$\pi_x = P(x) = F(x)/N$$

...the probability of  $x$  (unconditional).

# A transition matrix



$P(q_t|q_{t-1})$

	H	E	L
H	.93	.01	.06
E	.01	.80	.19
L	.04	.06	.90

\*\*This is a “first-order” MM. Transition probabilities depend on only the current state, not the state before it.

What is  $P(S|\lambda)$ , the probability of a sequence, given the model?

$$P(\text{"HHEELL"} | \lambda)$$

$$\begin{aligned} &= P(H)P(H|H)P(E|H)P(E|E)P(L|E)P(L|L) \\ &= (.33)(.93)(.01)(.80)(.19)(.90) \\ &= 4.2E-4 \end{aligned}$$

$\lambda$

	H	E	L
H	.93	.01	.06
E	.01	.80	.19
L	.04	.06	.90

$$P(\text{"HHHHHH"} | \lambda) = 0.69 \quad \leftarrow \text{common protein secondary structure}$$

$$P(\text{"HEHEHE"} | \lambda) = 1E-6 \quad \leftarrow \text{not protein secondary structure}$$

Probability discriminates between realistic and unrealistic sequences

# What is the *maximum likelihood* model given a dataset of sequences?

Dataset.

HHEELL



HHEELL

HHEELL

HHEELL

HHEELL

HHEELL



	H	E	L
H	1	1	0
E	0	1	1
L	0	0	1



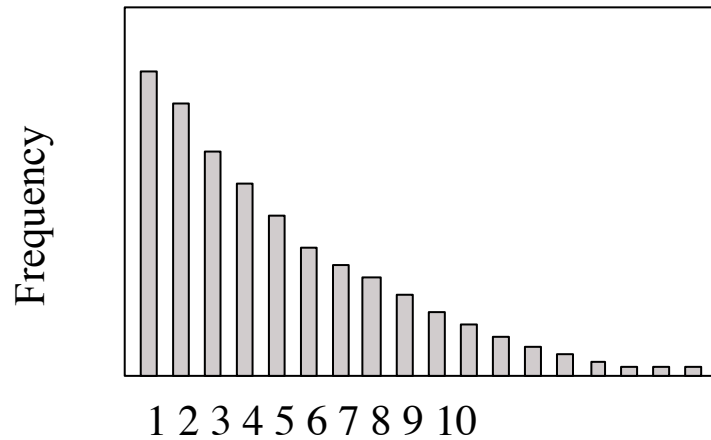
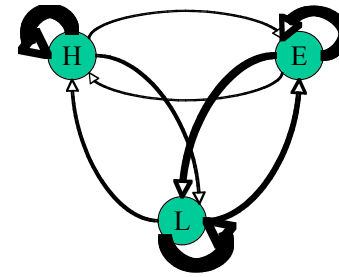
	H	E	L
H	0.5	0.5	0
E	0	0.5	0.5
L	.0	0	1.0

*Maximum likelihood model*

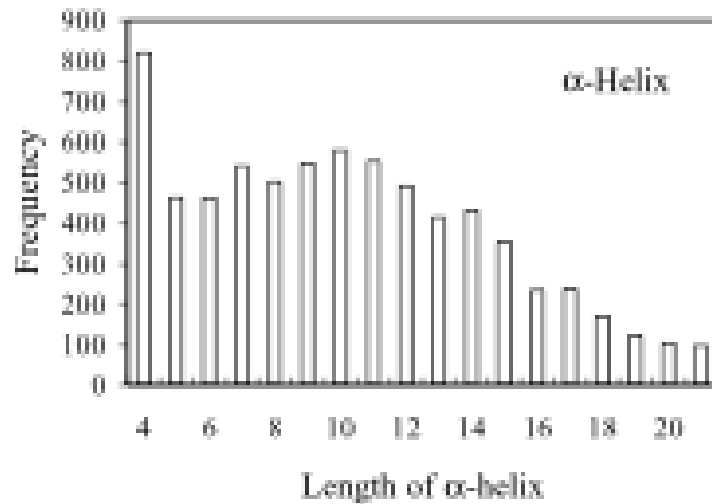
Count the state pairs.

Normalize by row.

Is this model too simple? →



Synthetic helix length data from this model



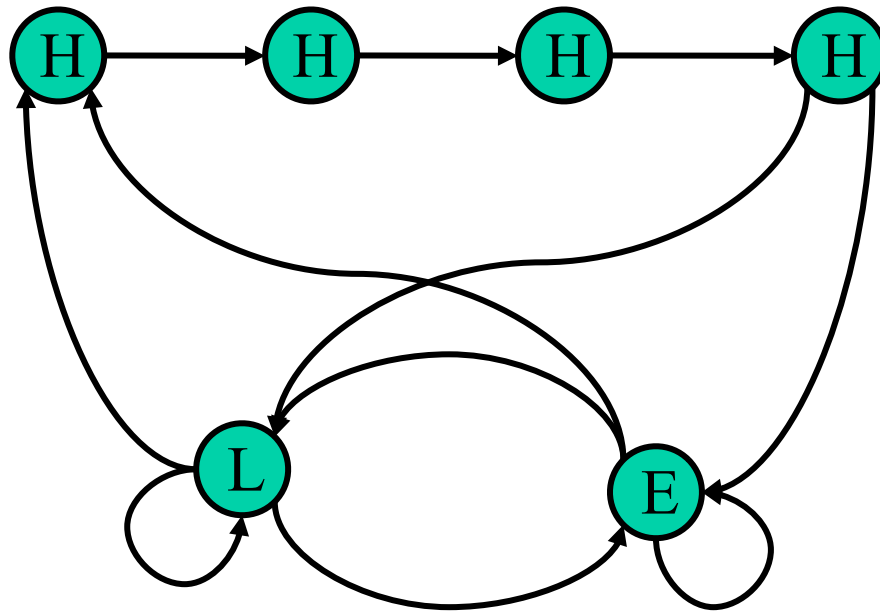
Real helix length data

\*L.Pal et al, J. Mol. Biol. (2003)  
326, 273–291

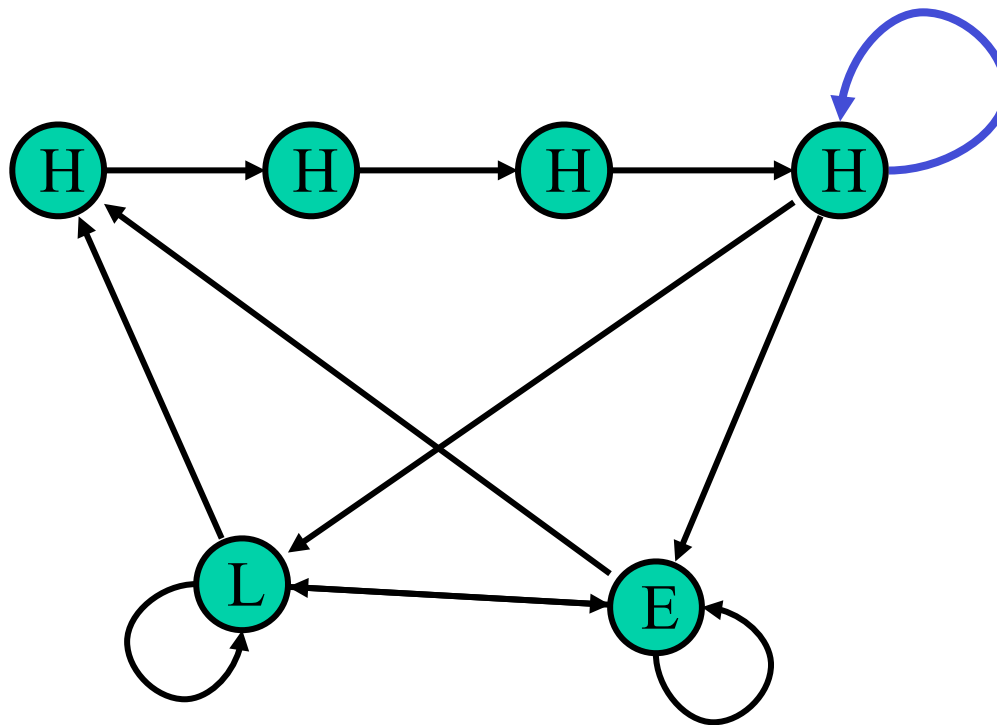
*“A model should be as simple as possible but not simpler” --Einstein*

# A pseudo-higher-order HMM

A Markov chain for proteins where helices are always exactly 4 residues long



A Markov chain for proteins where helices are always *at least* 4 residues long



Can you draw a Markov chain where helices are always a multiple of 4 long?

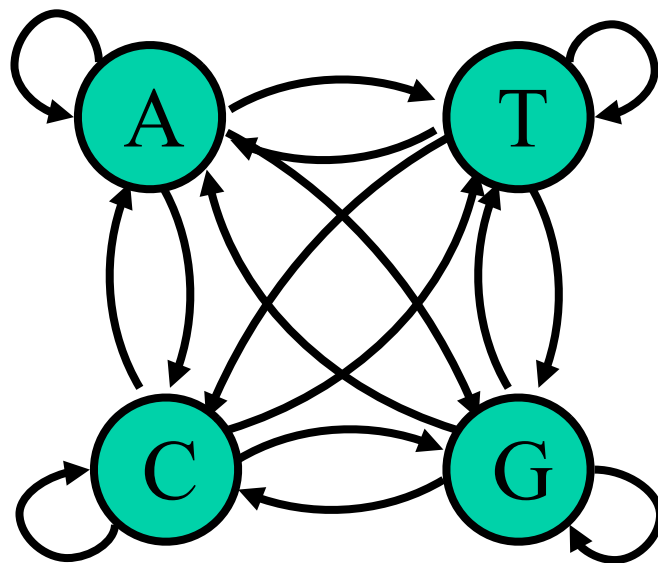
Exercise: generate a MM topology based on the data.

how much wood would a wood  
chuck chuck if a wood  
chuck would chuck wood?

can you can a can as a  
canner can can a can?

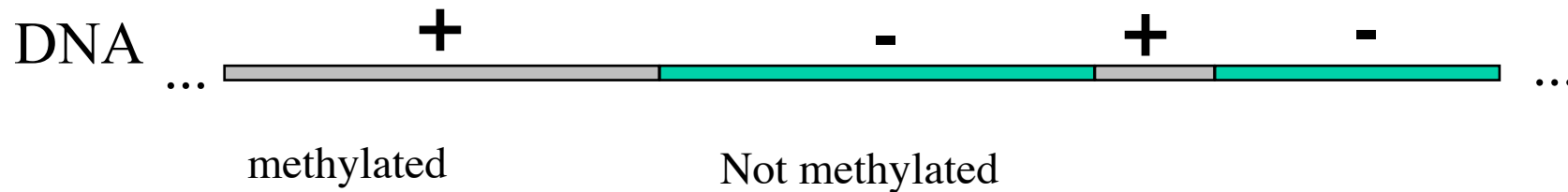
I wish to wish the wish  
you wish to wish, but if  
you wish the wish the  
witch wishes, I won't wish  
the wish you wish to wish.

## Application: A Markov chain for CpG islands



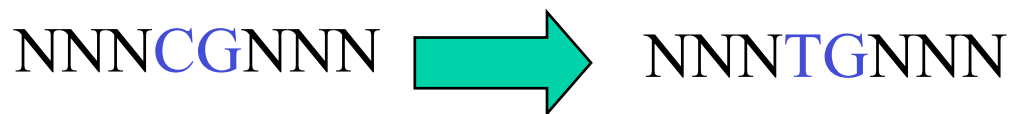
$$P(\text{ATCGCGTA}\dots) = \pi_A a_{AT} a_{TC} a_{CG} a_{GC} a_{CG} a_{GT} a_{TA} \dots$$

# CpG Islands



DNA is methylated on C to protect against endonucleases.

Using mass spectroscopy we can find regions of DNA that are methylated and regions that are not. Regions that are protected from methylation may be functionally important, i.e. transcription factor binding sites.



During the course of evolution. Methylated CpG's get mutated to TpG's



## Comparing two MMs

### The log likelihood ratio (LLR)

$$\log \prod_{i=1}^L \frac{a_{x_{i-1}x_i}^+}{a_{x_{i-1}x_i}^-} = \sum_{i=1}^L \log \frac{a_{x_{i-1}x_i}^+}{a_{x_{i-1}x_i}^-} = \sum_{i=1}^L \beta_{x_{i-1}x_i}$$

Log-likelihood ratios  
for transitions:

$\beta$	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

Sum the LLRs.

If the result is positive, its a CpG island, otherwise not.

$$\mathbf{LLR(CGCG)=1.812 + 0.461 + 1.812 = 4.085} \leftarrow \text{yes}$$

In class exercise: what's the LLR?

What is the LLR that this seq is a CpG Island?

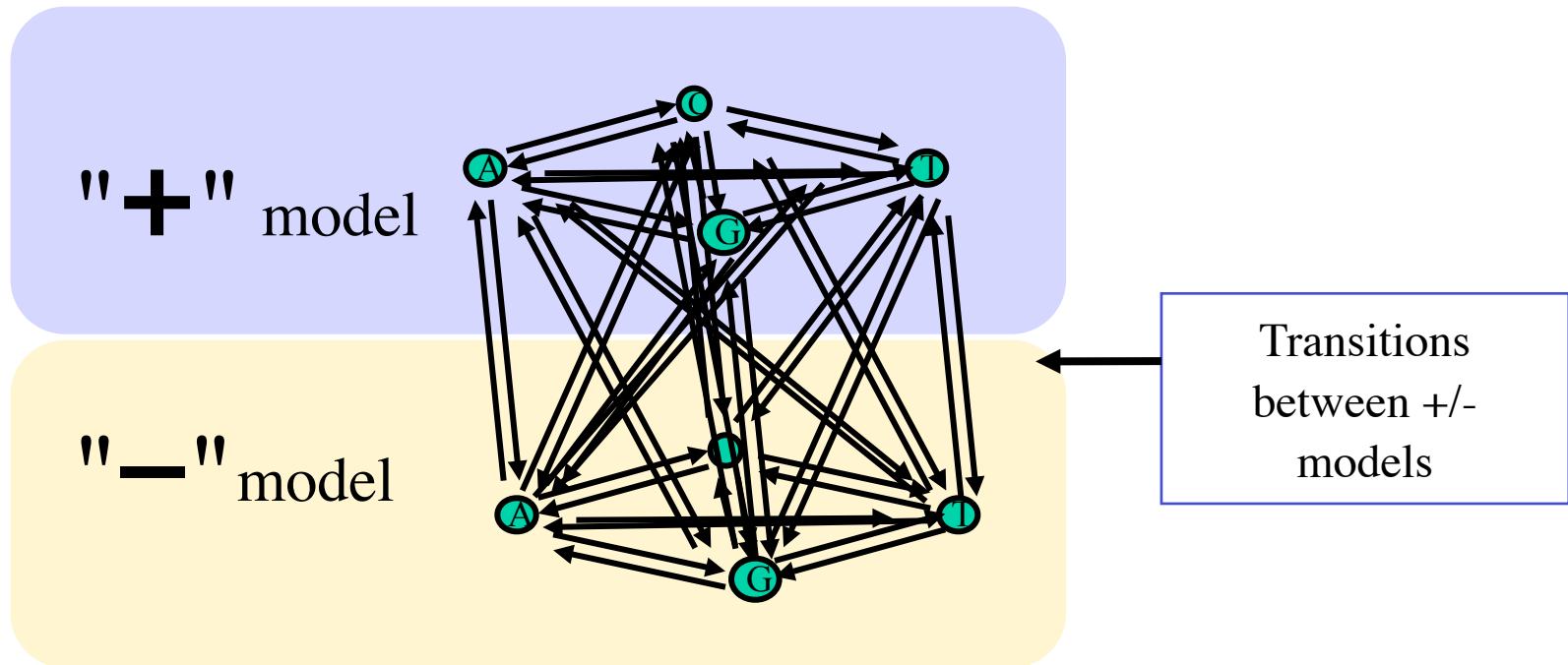
ATGTCTTAGCGCGATCAGCGAAAGCCACG

$\beta$	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

$$LLR = \sum_{i=1}^L \beta_{x_{i-1}x_i} = \underline{\hspace{10em}}$$

Combining two Markov chains to make a hidden Markov model

A *hidden* Markov model can have multiple paths for a sequence



In *Hidden Markov models* (HMM), there is no one-to-one correspondence between the state and the emitted symbol.

# Probability of a sequence using a HMM

*Different state sequences can produce the same emitted sequence*

**Nucleotide sequence: C G C G**

**State sequences (paths):**

**C+ G+ C+ G+**

**C- G- C- G-**

**C+ G+ C- G-**

**C+ G- C- G+**

**etc....**

P(sequence,path)

$$\pi_{C+} a_{C+G+} a_{G+C+} a_{C+G+}$$

$$\pi_{C-} a_{C-G-} a_{G-C-} a_{C-G-}$$

$$\pi_{C+} a_{C+G+} a_{G+C-} a_{C-G-}$$

$$\pi_{C+} a_{C+G-} a_{G-C-} a_{C-G+}$$

etc....

---

$$P(CGCG|\lambda) = \sum P(Q)$$

All paths Q

Each state sequence has a probability. The sum of all state sequences that emit CGCG is the P(CGCG).

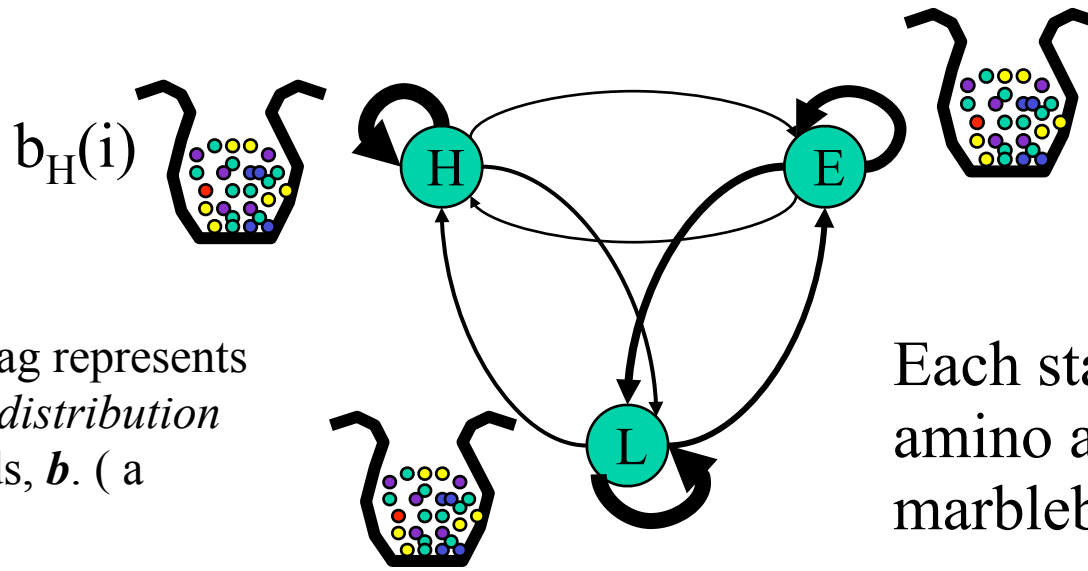
# HMM: assigning the states given the sequence is not as easy.

Typically, when using a HMM, the task is to determine the **optimal** state pathway given the sequence. The state pathway provides some *predictive feature*, such as secondary structure, or splice site/not splice site, or CpG island/not CpG island, etc.

**In Principle**, we can do this task by *trying all state pathways  $Q$ , and choosing the optimal*. **In Practice**, this is usually impossible, because the number of pathways increases as the number of states to the power of the length, *i.e.*  $O(n^m)$ .

How do we do it, then?

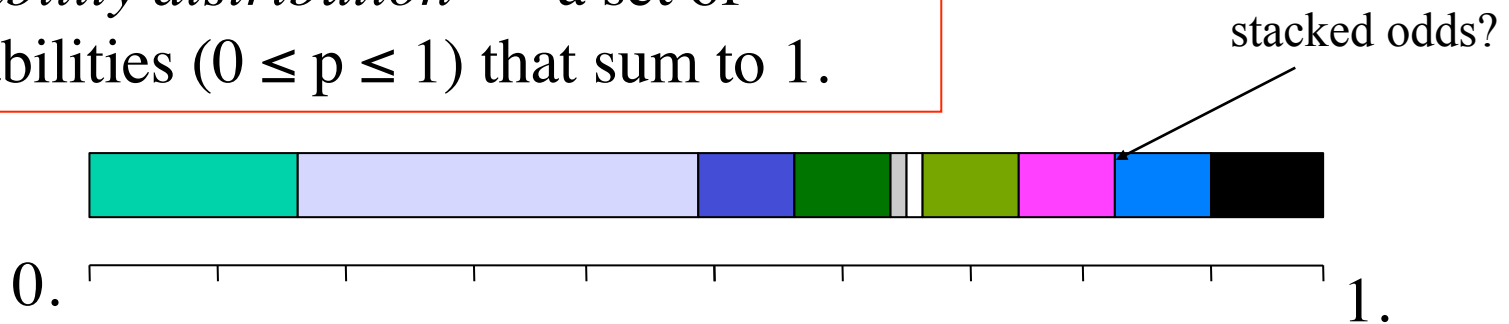
# Parallel HMM: emits sec struct *and* amino acid



The marble bag represents a *probability distribution* of amino acids,  $\mathbf{b}$ . ( a profile )

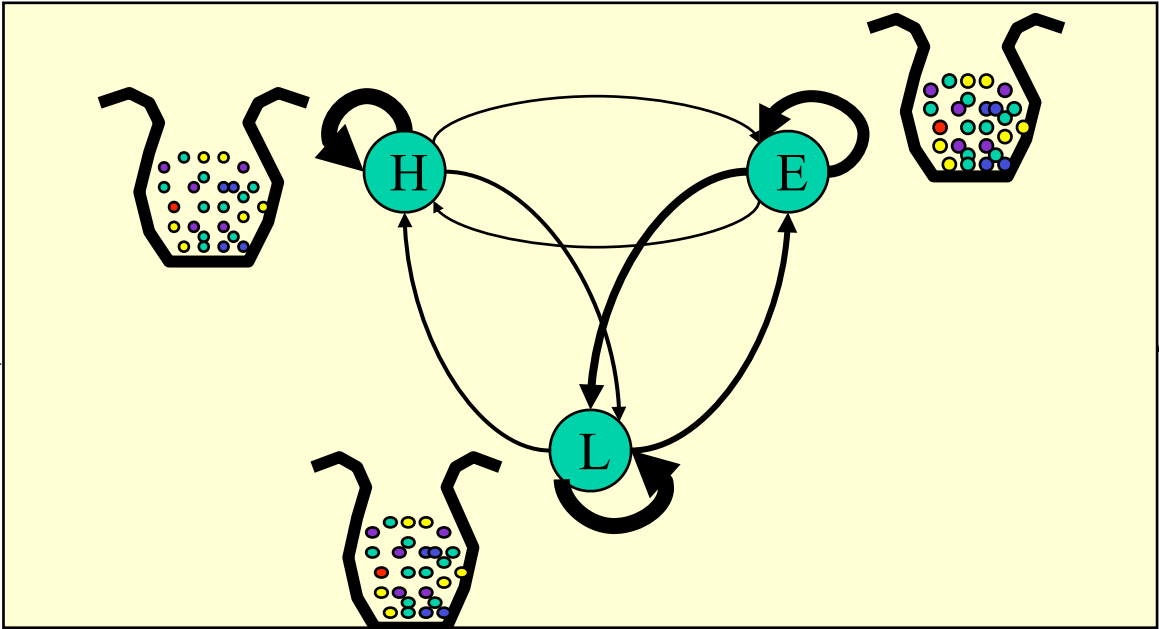
Each state emits one amino acid from the marblebag, for each visit.

*probability distribution* == a set of probabilities ( $0 \leq p \leq 1$ ) that sum to 1.



states emit aa and ss.

Amino acid  
Sequence



State sequence  
(secondary  
structure)

*Given an amino acid sequence, what is the most probable state sequence?*

Maximize:

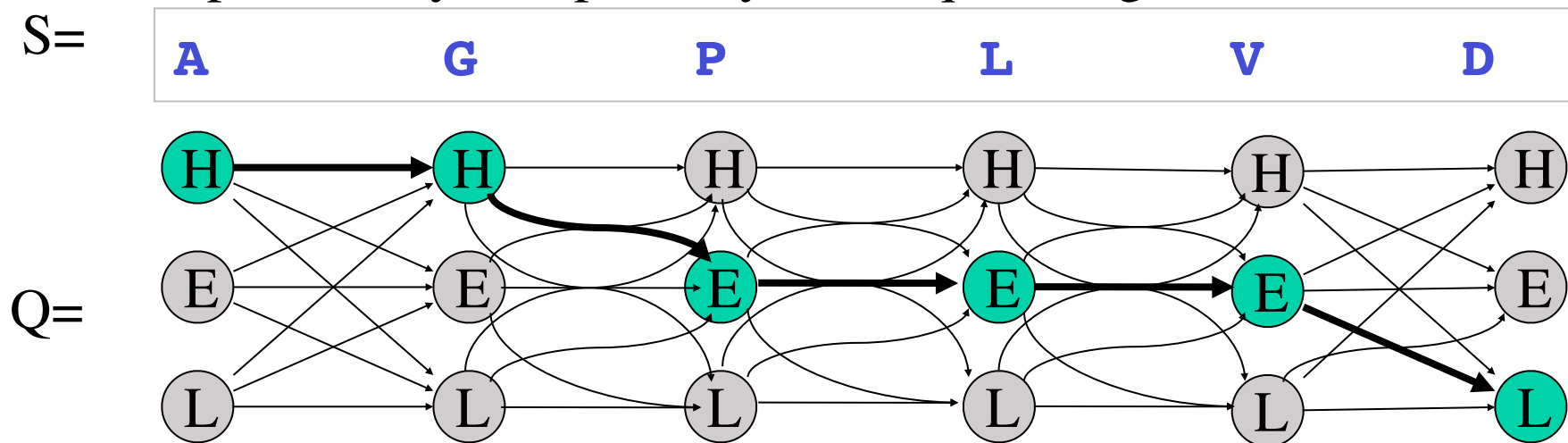
## Joint probability of a sequence and pathway

$Q = \{q_1, q_2, q_3, \dots, q_T\}$  = sequence of Markov states, or pathway

$S = \{s_1, s_2, s_3, \dots, s_T\}$  = sequence of amino acids or nucleotides

$T$  = length of  $S$  and  $Q$ .

Joint probability of a pathway and sequence, given a HMM  $\lambda$ .



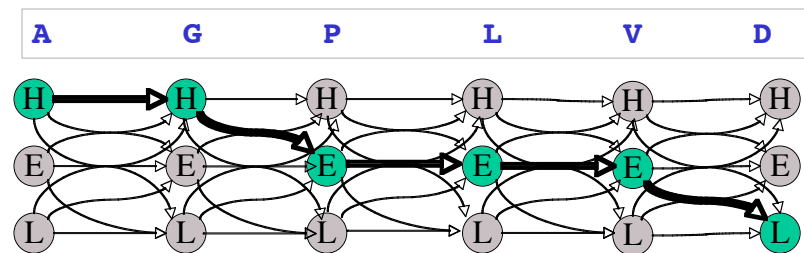
$$P = \pi_H b_H(\mathbf{A}) \times a_{HH} b_H(\mathbf{G}) \times a_{HE} b_E(\mathbf{P}) \times a_{EE} b_E(\mathbf{L}) \times a_{EE} b_E(\mathbf{V}) \times a_{EL} b_L(\mathbf{D})$$

# Joint probability : general expression

General expression for pathway  $Q$  through HMM  $\lambda$  :

$$P(S, Q | \lambda) = \pi_{q_1} \prod_{t=1, T} b_{q_t}(s_t) a_{q_t q_{t+1}} \quad **$$

\*\*when  $t=T$ , there is no  $q_{t+1}$ . Use  $a = 1$



# The Three HMM Algorithms

1. The **Viterbi** algorithm: get the optimal state pathway.  
Maximum joint prob.
2. The **Forward/Backward** algorithm: get the probability of each state at each position. Sum over all joint probs.
3. **Expectation/Maximization**: refine the parameters of the model using the data

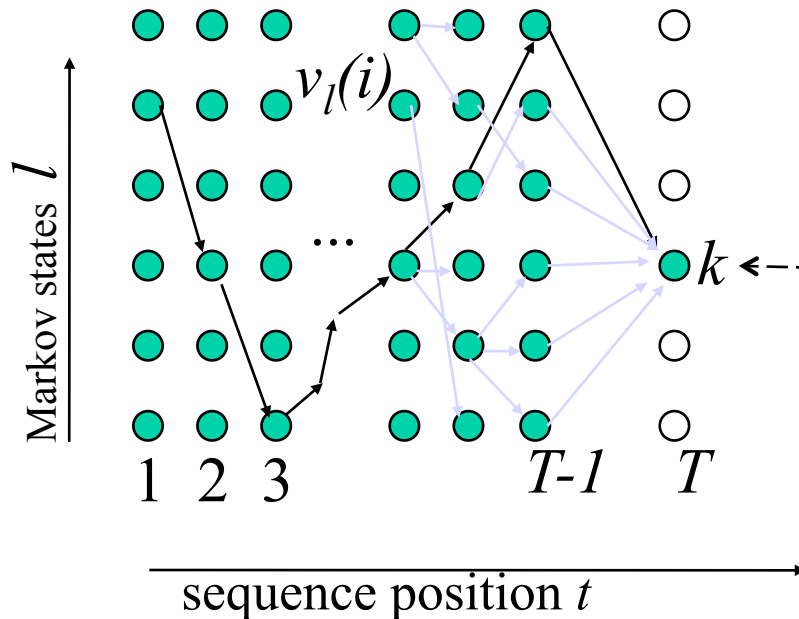
# The Viterbi algorithm: the maximum probability path

$$v_k(t) = \text{MAX}_l v_l(t-1) a_{lk} b_k(s_t)$$

$$\text{Trc}_k(t) = \underset{l}{\text{ARGMAX}} v_l(t-1) a_{lk} b_k(s_t)$$

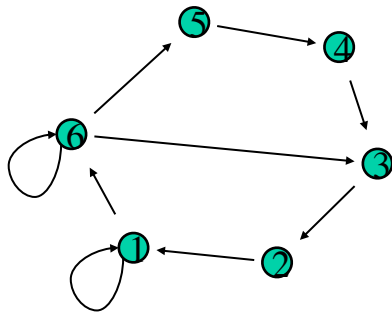
Recursive. We save the value  $v$  and also a traceback arrow  $\text{Trc}$  as we go along.

Plot state versus position. Each  $v$  is a MAX over the whole previous column of  $v$ 's.



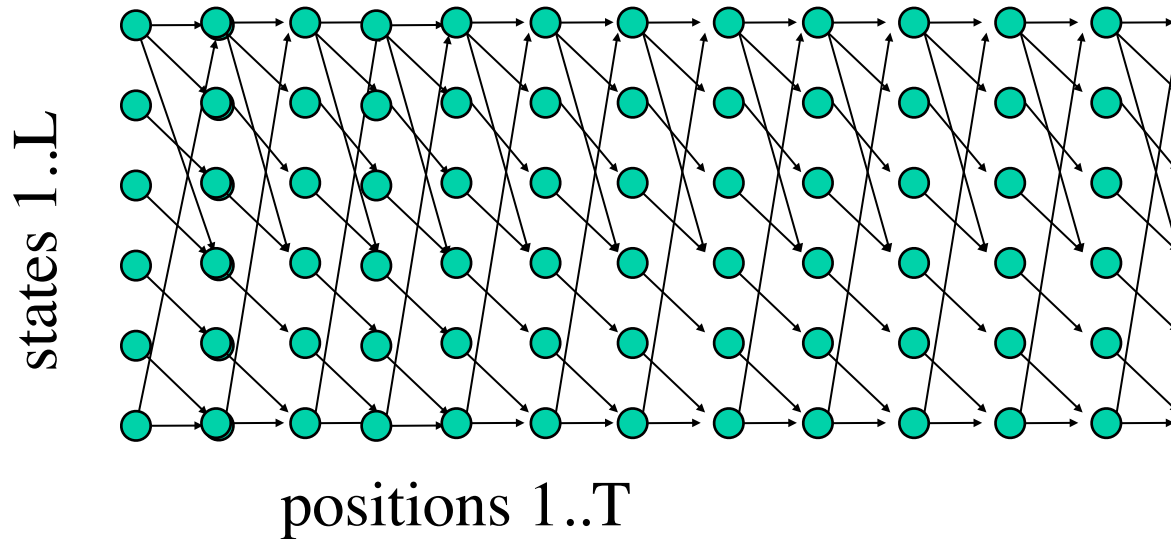
When  $t = T$  the last position, the traceback arrow from the MAX give the optimal state sequence.

## Exercise: Write the Viterbi algorithm

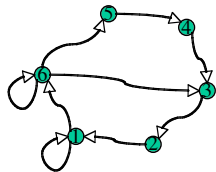


$$v_k(t) = \text{MAX } v_l(t-1) a_{lk} b_k(s_t)$$

$$\text{Trc}_k(t) = \text{ARGMAX } v_l(t-1) a_{lk} b_k(s_t)$$

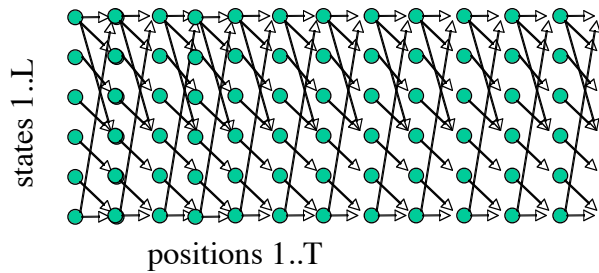


## Exercise: Write the Viterbi algorithm



$$v_k(t) = \text{MAX } v_l(t-1) a_{lk} b_k(s_t)$$

$$\text{Trc}_k(t) = \text{ARGMAX } v_l(t-1) a_{lk} b_k(s_t)$$



initialize  $v_k(1)=b_k(s_1)$

for  $t=2,T$  {

    for  $k=1,L$  {

    }

}

# The Forward algorithm: all paths to a state

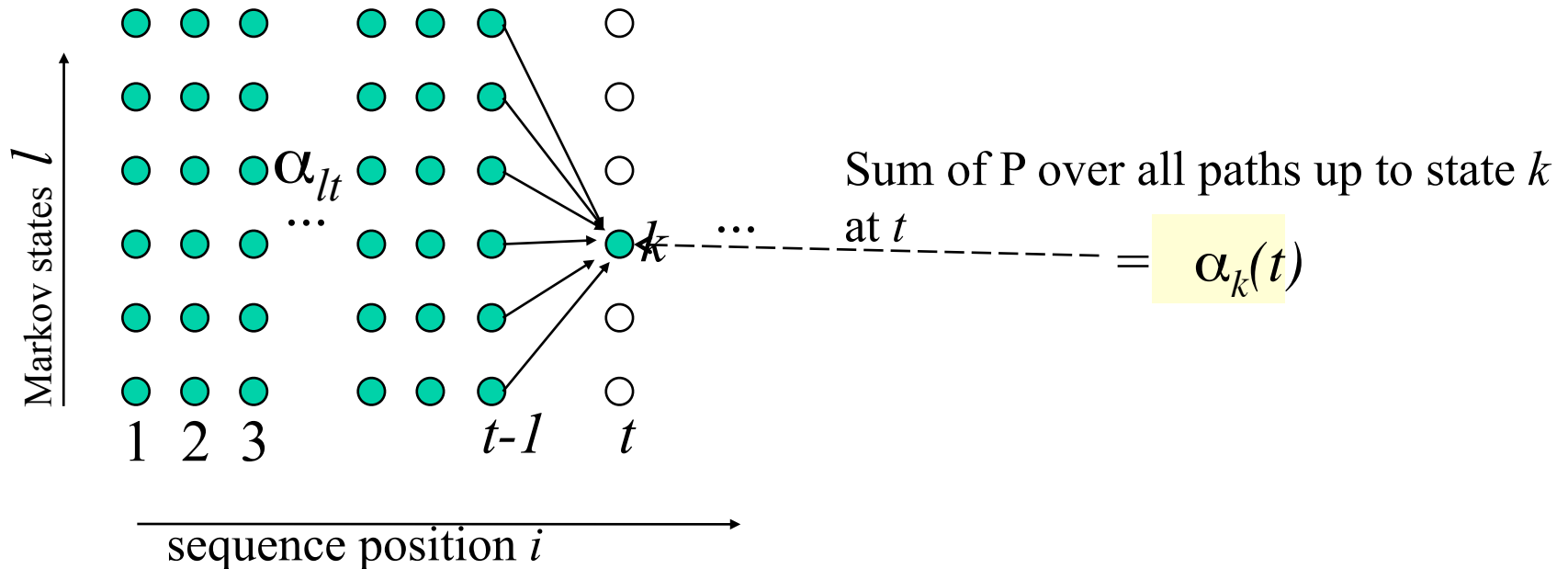
This is *alpha*, the forward probability

This is 'a', the 'arrow' between states.

$$\alpha_k(t) = \sum_l \alpha_l(t-1) a_{lk} b_k(t)$$

“Forward” stands for “forward recursion”

After the first row, each  $\alpha$  depends on the whole previous row of  $\alpha$ 's.



At the end of the sequence, when  $t=T$ , the sum of  $\alpha_k(T)$  equals the total probability of the sequence given the model,  $P(S|\lambda)$ .

## The Backward algorithm: all paths from a state

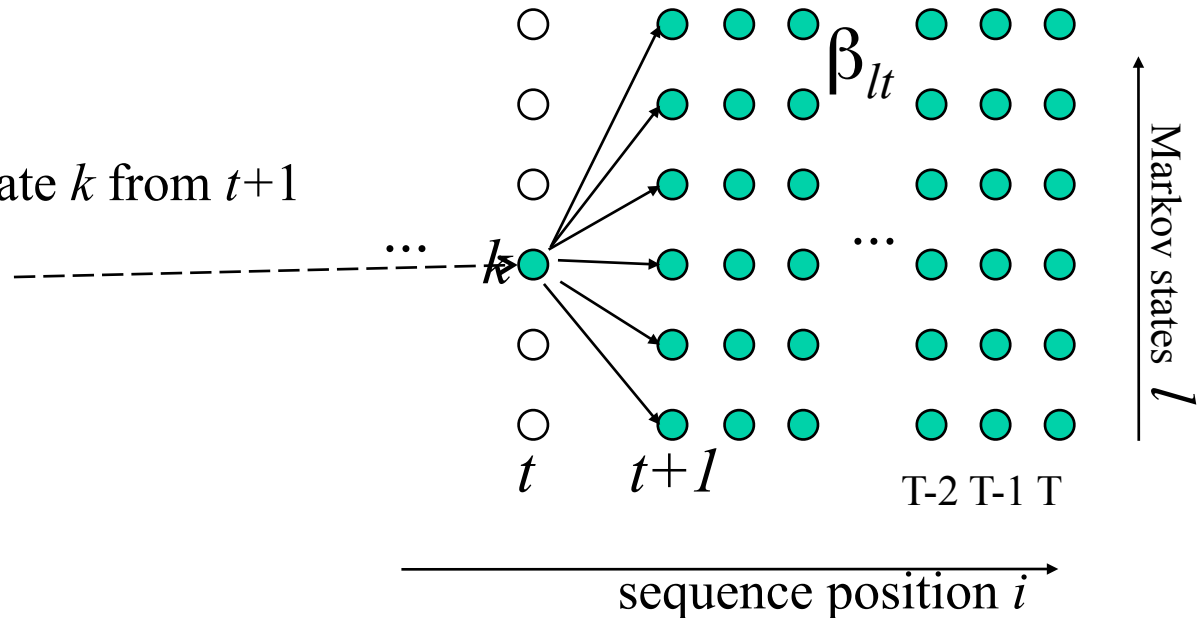
$$\beta_k(t) = \sum_l \beta_l(t+1) a_{kl} b_k(t)$$

“Backward” stands for “backward recursion”. The algorithm starts at  $t=T$ , the end of the sequence. (The transitions are still forward.)

Each  $\beta$  depends on the whole *next* row of  $\beta$ 's.

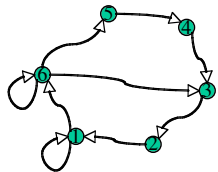
Sum over all paths to state  $k$  from  $t+1$

$$= \beta_k(t)$$

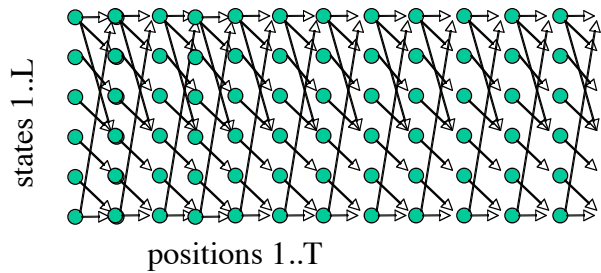


At the beginning of the sequence, when  $t=1$ , the sum of  $\beta_k(1)$  equals the total probability of the sequence given the model,  $P(S|\lambda)$ .

## Exercise: Write the Forward algorithm



$$\alpha_k(t) = \sum_j \alpha_j(t-1) a_{jk} b_k(t)$$



initialize  $\alpha_k(1) = \pi_k(s_1)$

for  $t=2, T$  {

    for  $k=1, L$  {

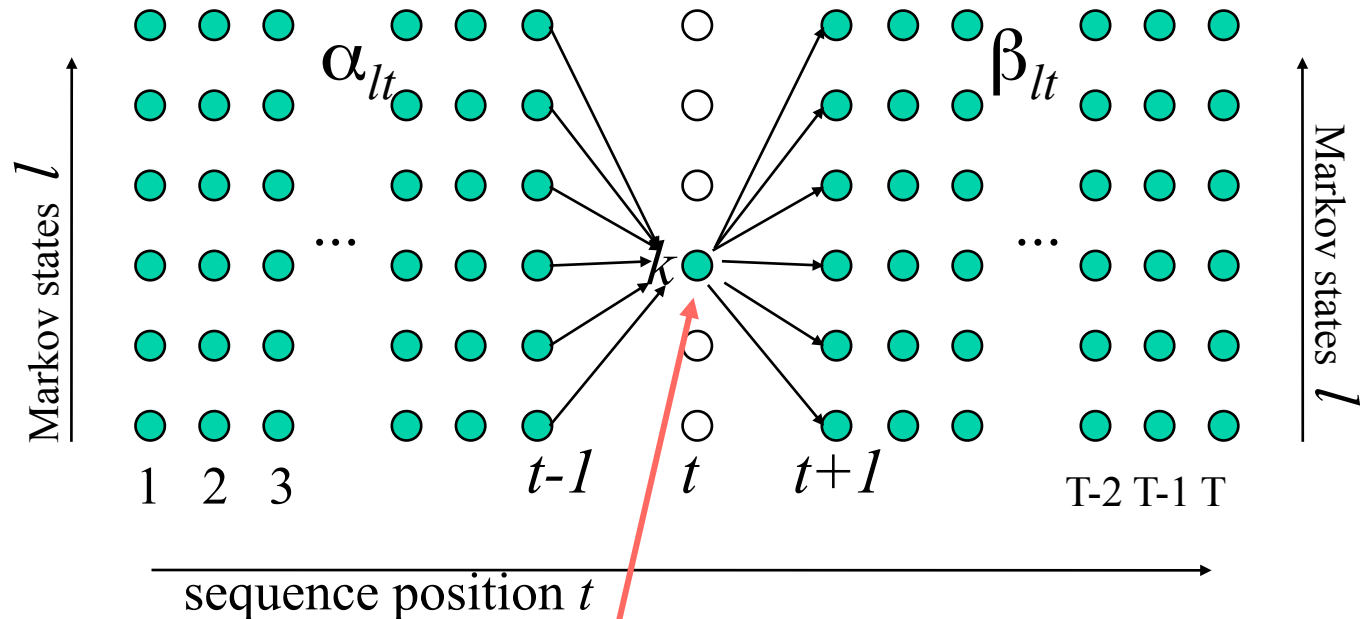
    }

}

**Forward/Backward algorithm**: all paths through a state.

$$\gamma_k(t) = \alpha_k(t) * \beta_k(t)$$

$\gamma_k(t)$  is the total probability of state  $k$  at  $t$ , given the sequence  $S$  and the model,  $\lambda$ .



The bottleneck through which all paths must travel.

## Expectation/Maximization: refining the model

Example: refining  $b_k(\text{G})$  (i.e. the number of Gly's in the  $k^{\text{th}}$  marble bag)

Step 1) Count how many Glycines are found in state  $k$ .

Step 2) Normalize it. Reset  $b_k(\text{G})$  in the new model to that value.

Step 3) Do steps 1-2 for all states  $k$  in  $\lambda$  and all 20 amino acids.

Repeat steps 1-3 using the new model. Iterate to convergence.

Expectation/Maximization is often abbreviated “EM”.

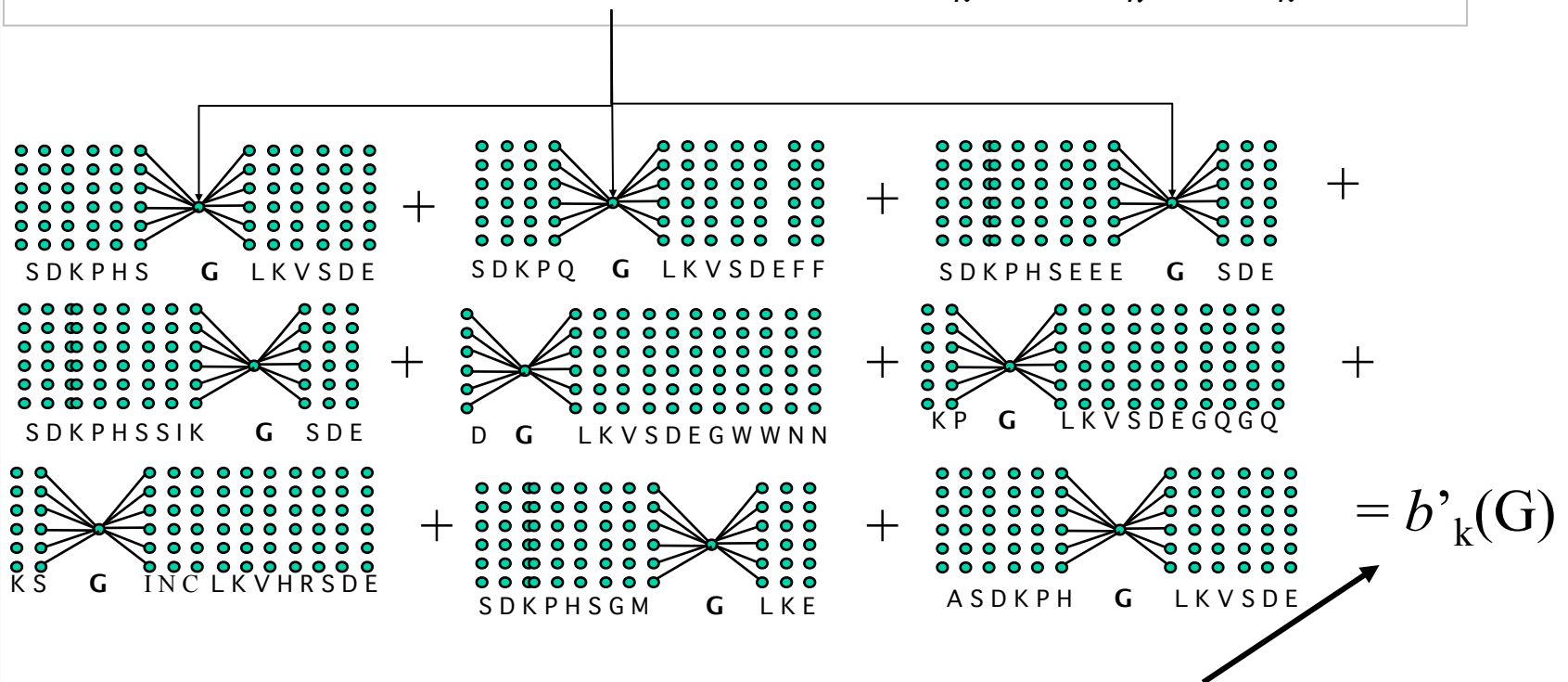
# Expectation/Maximization: refining the model

## Example: refining $b_k(G)$

To count the Glycines, we calculate the Forward/Backward value for state  $k$  at every Glycine in the database. Then sum them.

$$P(k|t, S, \lambda) = \sum \text{all paths through } k \text{ at } t = \gamma_k(t) = \alpha_k(t) * \beta_k(t)$$

Σ over all G in all sequences, S



This is normalized to sum to 1 over all 20 AA's.

## Expectation/Maximization: refining the model

Example: refining  $a_{jk}$ , the probability of a transition from state  $j$  to state  $k$ .

Step 1) Get the probability of ending in state  $j$  at  $t$   
-->  $\alpha_j(t)$

Step 2) Get the probability of starting in state  $k$  at  $t+1$   
-->  $\beta_k(t)$

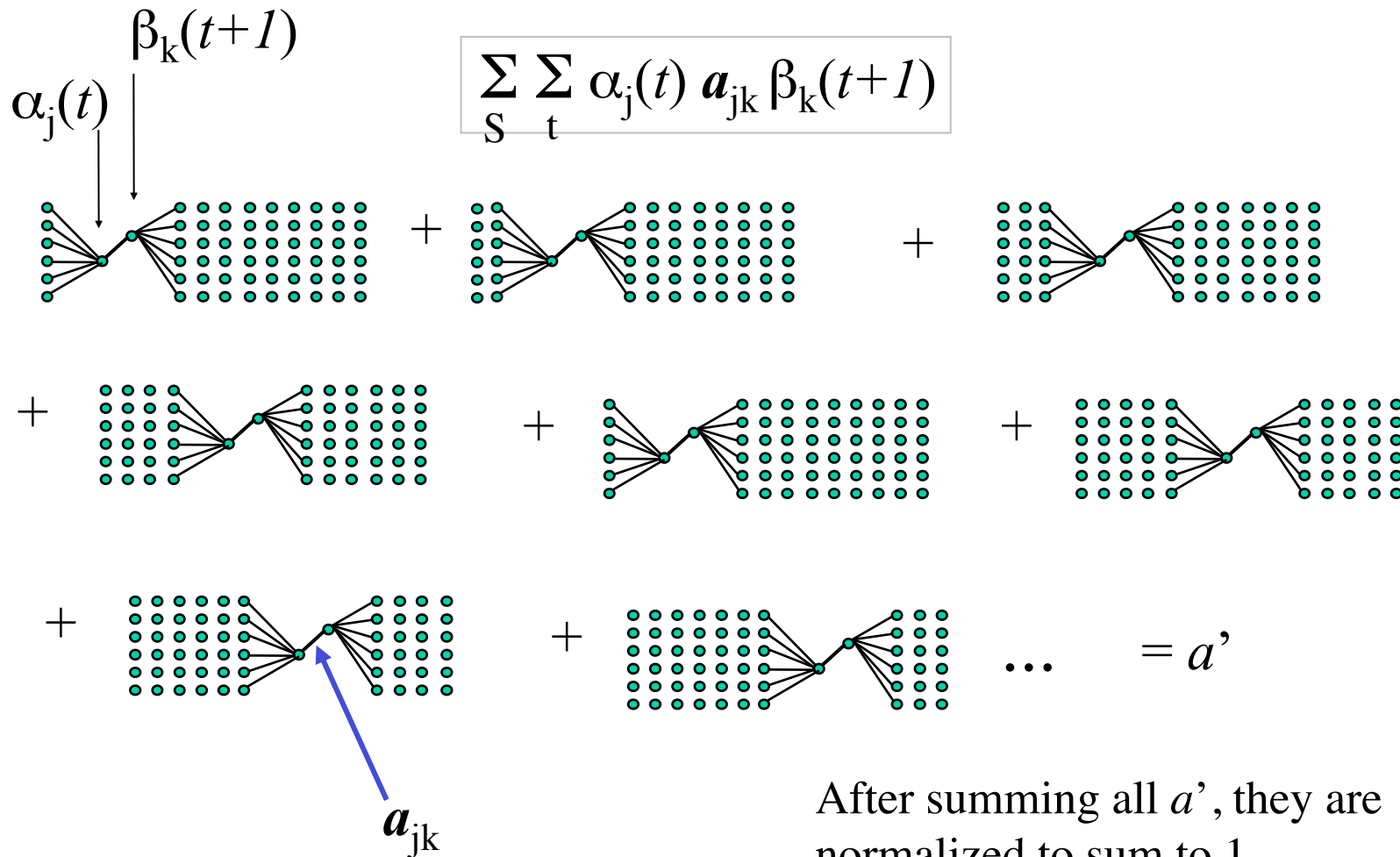
Step 3) Multiply these by the current  $a_{jk}$

Step 4) Do Steps 1-3 for all positions  $t$  and all sequences,  $S$ .  
Sum-->  $a'$ . Then normalize. Reset  $a_{jk}$  in the new model to  $a'$ .

Do 1-4 using the new model. Repeat until convergence.

# Expectation/Maximization: refining the model

Example: refining  $a_{jk}$ , the probability of a transition from state  $j$  to state  $k$ .



$\Sigma$  over all  $t$  in all sequences,  $S$

After summing all  $a'$ , they are normalized to sum to 1.

# “Profile HMMs”

## State emissions:

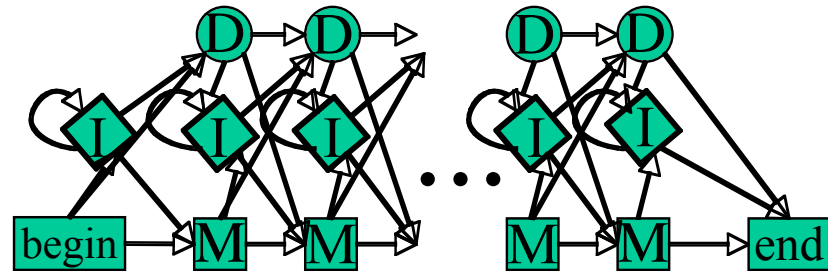
I = insert state, one character from the background profile

D = delete state, non-emitting. A connector.

M = match state, one character from a specific profile.

Begin = non-emitting. Source state.

End = non-emitting. Sink state.



All  $\pi(q)=0$ , except  $\pi(\text{Begin})=1$

To get the scores of a sequence to a profile HMM, we use the F/B algorithm to get  $P(\text{End})$ . This is the measure of how well the sequence fits the model.

Then we can test several models.

# Generating a profile HMM from a multiple sequence alignment

base the model on, say, this one

**VGA--H**

**V-----N**

**VEA--D**

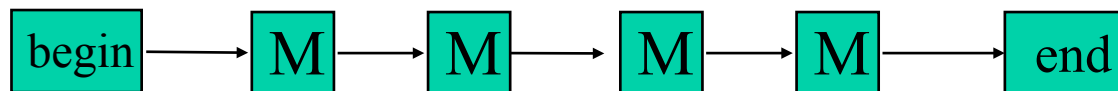
**VKG---**

**VYS--T**

**FNA--N**

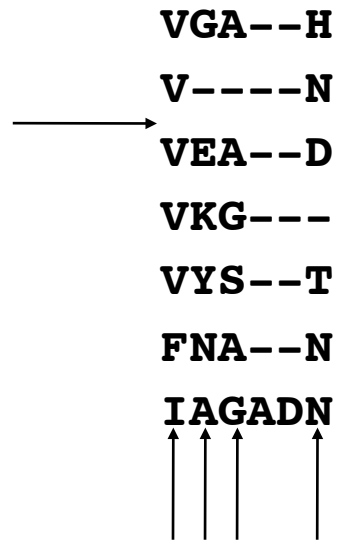
**IAGADN**

Make four match states



# Generating a profile HMM from a multiple sequence alignment

base the  
model on, say,  
this one



Make four  
match states

# Generating a profile HMM from a multiple sequence alignment

VGA--H

V-----N

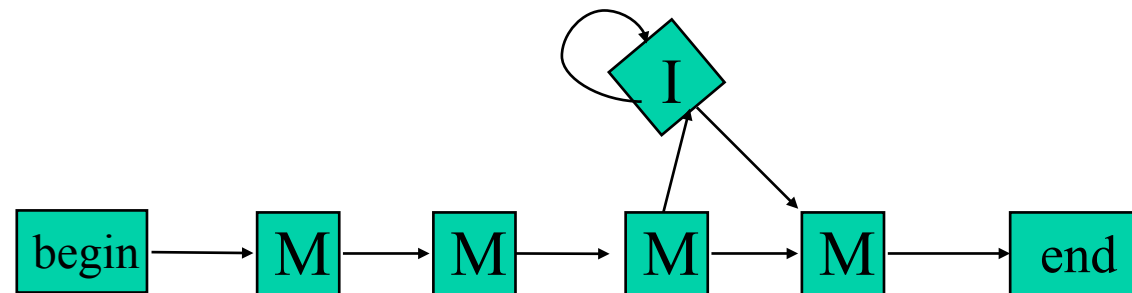
VEA--D

VKG----

VYS--T

FNA--N

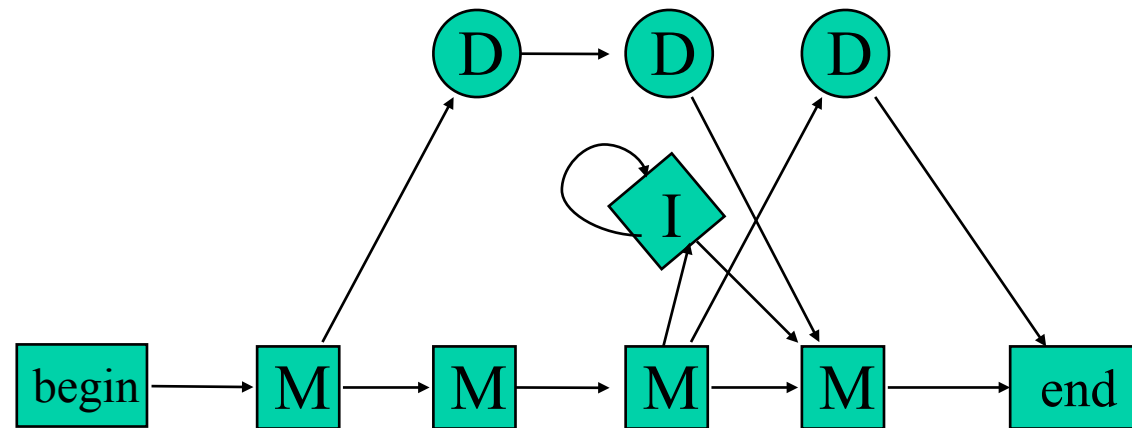
IAGADN



Add insertion states where there are insertions.  
(red)

# Generating a profile HMM from a multiple sequence alignment

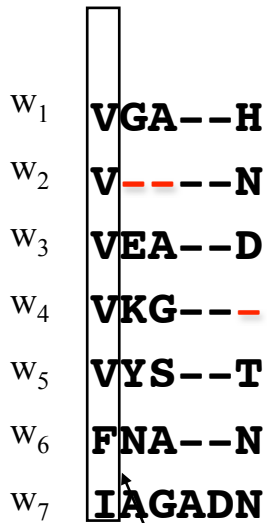
VGA--H  
V-----N  
VEA--D  
VKG---  
VYS--T  
FNA--N  
IAGADN



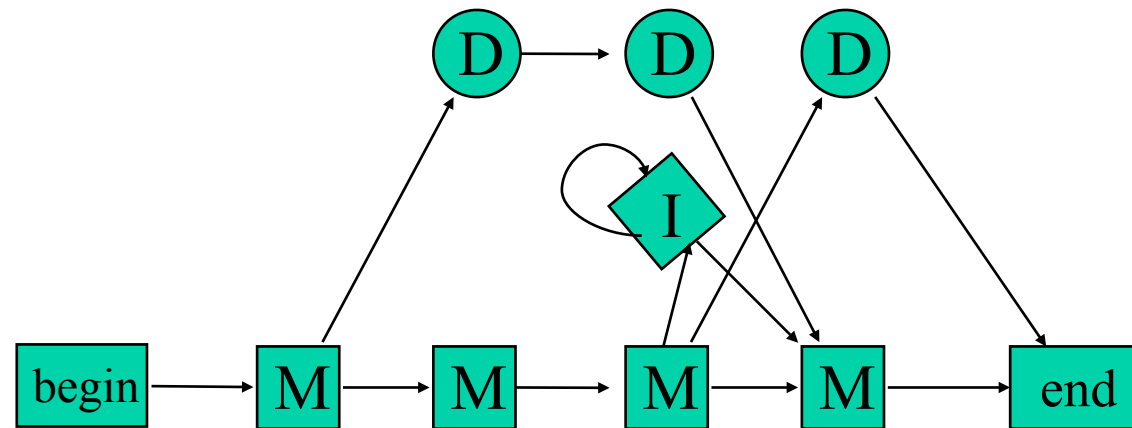
Add deletion states where there are deletions. (red dashes)

...now optimize using *expectation maximization*.

# Getting profiles for every Match state



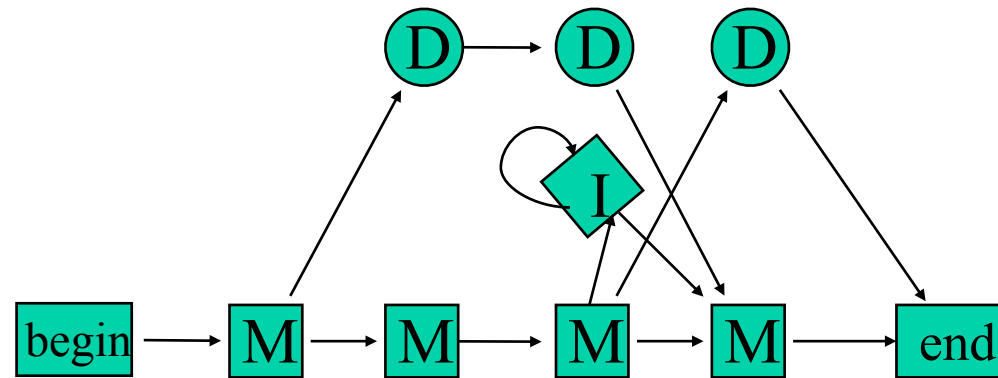
Count the frequency of each amino acid, scaled by sequence weights,  $w$ .



$$P(V) = \frac{\sum_{s_i=V} w_i}{\sum_{all\ i} w_i}$$

$$b_{M1}(V) = (w_1 + w_2 + w_3 + w_4 + w_5) / (w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7)$$

Calculating the probability of a sequence  
given the model:  $P(s|\lambda)$



**Sum forward (forward algorithm) using the sequence  $s$ .**

For each Match state, multiply by the transition ( $a$ ) and the profile value,  $b_M(s_i)$ , and increment  $i$

For each Deletion state, multiply by  $a$ , do not increment  $i$ .

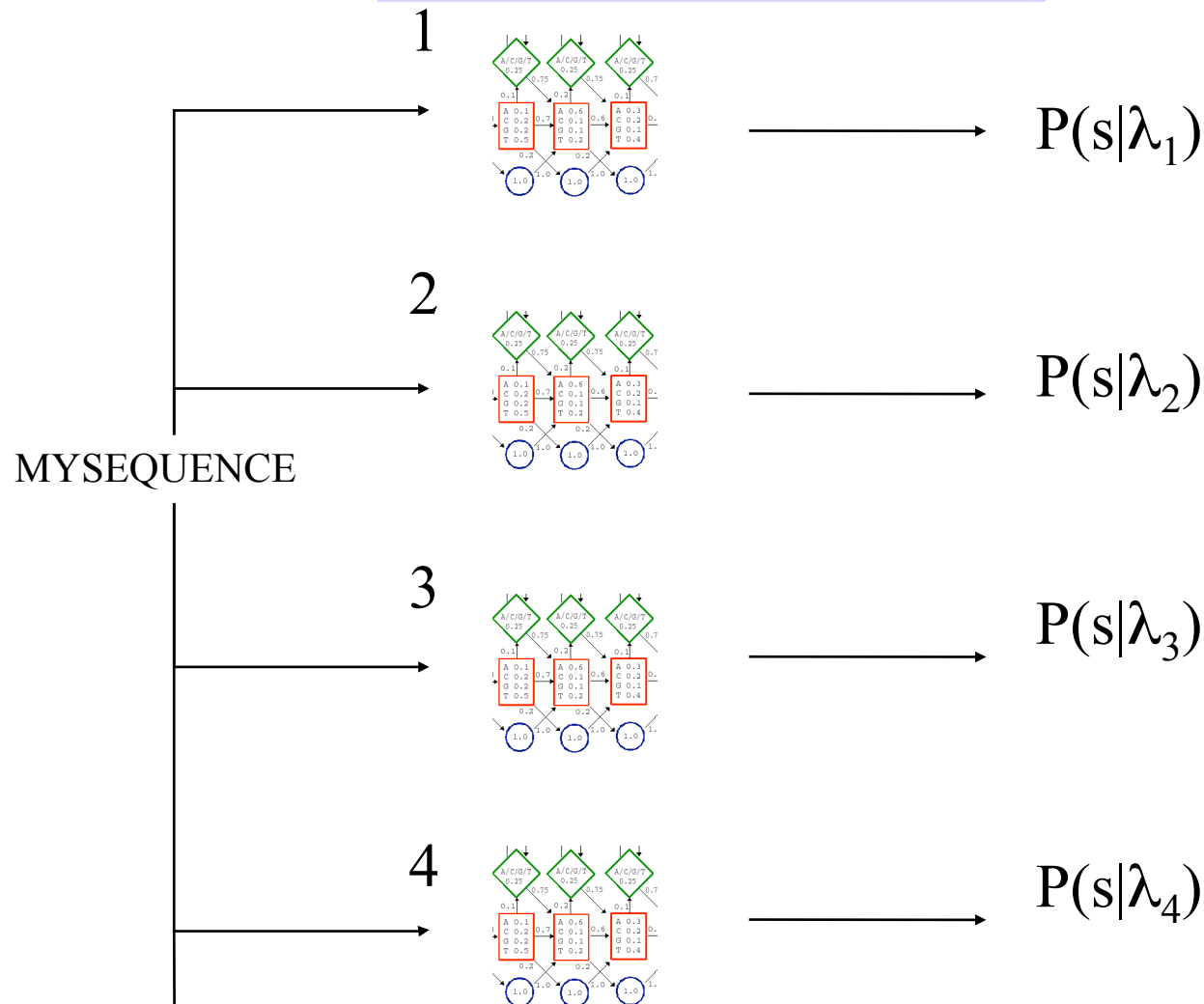
For each Insertion state, multiply by  $a$ , increment  $i$ .

# Picking a parent sequence

- The parent defines the number of Match states
- A Match state should conserve the *chemical nature of the sidechain* as much as possible.
- A Match state implies *structural similarity*.

# Homolog detection using a library of profile HMMs

Get  $P(S|\lambda)$  for each  $\lambda$



Pick the model with the max P

# In Class exercise: make a profile HMM

**AGF---PDG**

**AGGYL-PDG**

**AG-----PNG**

**SGFFLIPNG**

**SGF--EPNG**

- Pick the best parent. Draw match states.
- Draw insertion states for positions followed by "-" in the parent.
- Draw deletion states for positions in parent that align with "-".
- For each Match state, write the predominant amino acid.

# Make a HMM from Blast data

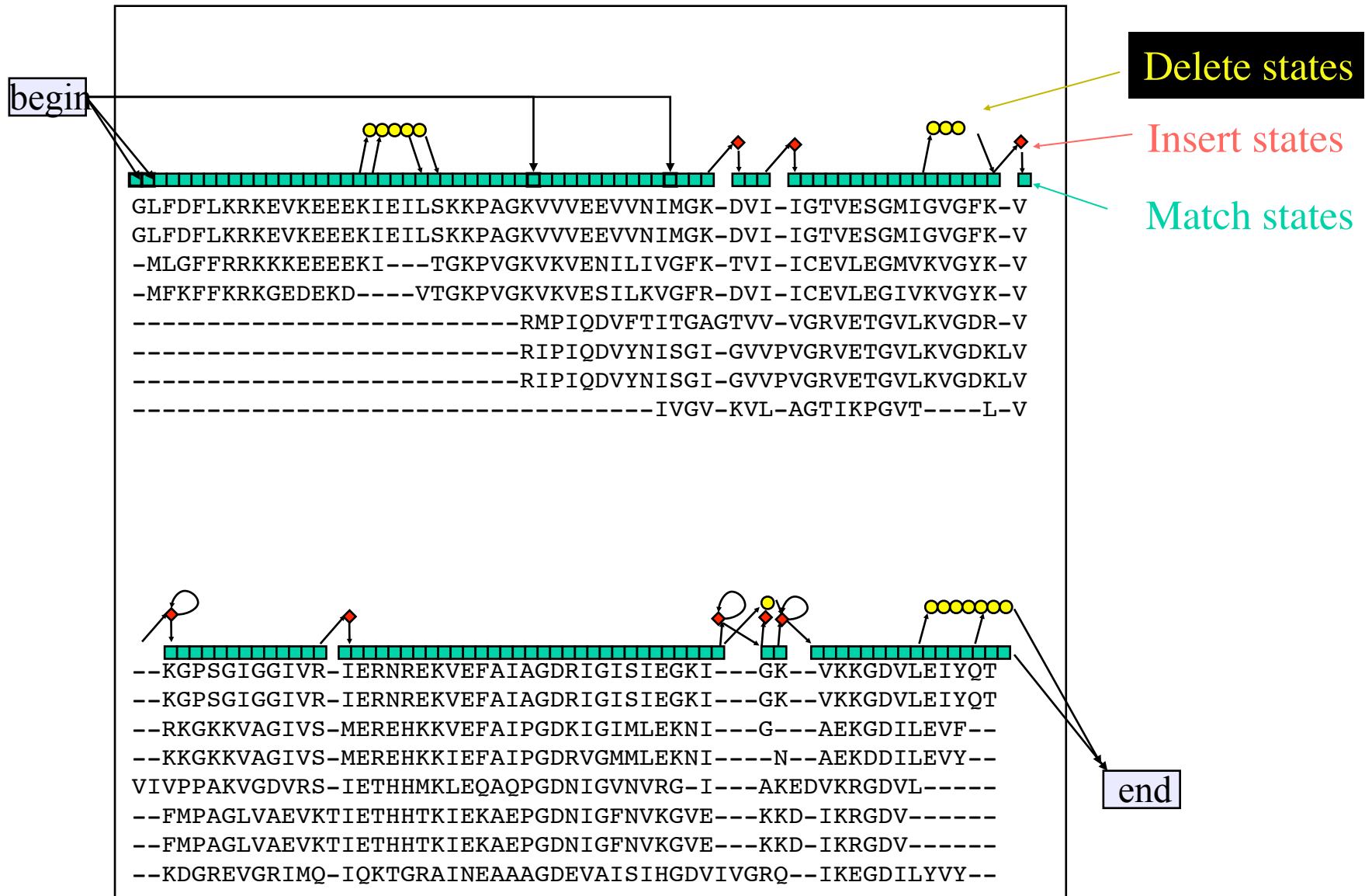
```
Score      E
Sequences producing significant alignments:                      (bits) Value

gi|18977279|ref|NP_578636.1| (NC_003413) hypothetical protein [P... 136 5e-32
gi|14521217|ref|NP_126692.1| (NC_000868) hypothetical protein [P... 59 8e-09
gi|14591052|ref|NP_143127.1| (NC_000961) hypothetical protein [P... 56 8e-08
gi|18313751|ref|NP_560418.1| (NC_003364) translation elongation ... 42 9e-04
gi|729396|sp|P41203|EF1A_DESMO Elongation factor 1-alpha (EF-1-a... 40 0.007
gi|1361925|pir||S54734 translation elongation factor aEF-1 alpha... 39 0.008
gi|18312680|ref|NP_559347.1| (NC_003364) translation initiation ... 37 0.060
```

```
QUERY      3  GLFDFLKRKEVKKEEKIEILSKKPAGKVVVEEVVNIMGK-DVI-IGTVESGMIGVGFK-V 59
18977279  2  GLFDFLKRKEVKKEEKIEILSKKPAGKVVVEEVVNIMGK-DVI-IGTVESGMIGVGFK-V 58
14521217  1  -MLGFFRRKKKEEEEKI---TGKPVGKVKVENILIVGFK-TVI-ICEVLEGMVKVGYK-V 53
14591052  1  -MFKFFKRKGEDEKD----VTGKPVGKVKVESILKVGFR-DVI-ICEVLEGIVKVGYK-V 52
18313751 243 -----RMPIQDVFTITGAGTVV-VGRVETGVLKVGDR-V 274
729396   236 -----RIPIQDVYNISGI-GVVPVGRVETGVLKVGDKLV 268
1361925  239 -----RIPIQDVYNISGI-GVVPVGRVETGVLKVGDKLV 271
18312680 487 -----IVGV-KVL-AGTIKPGVT----L-V 504
```

```
QUERY      60  --KGPSGIGGIVR-IERNREKVEFAIAGDRIGISIEGKI---GK--VKKGDVLEIYQT 109
18977279  59  --KGPSGIGGIVR-IERNREKVEFAIAGDRIGISIEGKI---GK--VKKGDVLEIYQT 108
14521217  54  --RKGKKVAGIVS-MEREHKKVEFAIPGDKIGIMLEKNI---G---AEKGDILEVF-- 100
14591052  53  --KKGKKVAGIVS-MEREHKKIEFAIPGDRVGMMLLEKNI----N--AEKDDILEVY-- 99
18313751 275  VIVPPAKVGDVRS-IETHMKLEQAQPGDNIGVNVRG-I---AKEDVKRGDVL----- 322
729396   269  --FMPAGLVAEVKTIETHHTKIEKAEPGDNIGFNVKGV---KKD-IKRGDV----- 314
1361925  272  --FMPAGLVAEVKTIETHHTKIEKAEPGDNIGFNVKGV---KKD-IKRGDV----- 317
18312680 505  --KDGREVGRIMQ-IQKTGRAINEAAAGDEVAISIHGDVIVGRQ--IKEGDILYVY-- 555
```

# Make a HMM from Blast data



# Added information

In DP, we assumed insertions and deletions were equally probable, and that the *probability was independent of position*.

With Profile HMMs we allow *insertions* and *deletions* to have different probabilities, and to be *dependent on the position*.

# Many uses of HMMs

Weather prediction

Ecosystem modeling

Brain activity

Language structure

Econometrics

etc etc

HMMs can be applied to any dataset that can be represented as strings.

The expert input is the “topology”, or how the states are connected.

# Profile HMM libraries available via web

Pfam (HMMer):

[pfam.wustl.edu](http://pfam.wustl.edu)

SAM:

[www.cse.ucsc.edu/research/compbio/HMM-apps/](http://www.cse.ucsc.edu/research/compbio/HMM-apps/)