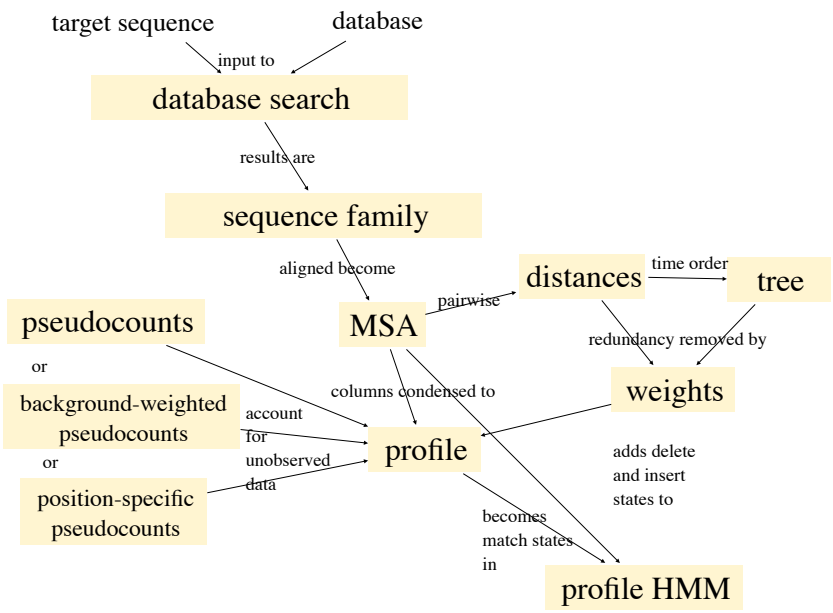
Bioinformatics 1--lectures 15, 16

Markov chains

Hidden Markov models

Profile HMMs


target sequence
database
input to
database search
results are
sequence family
aligned become
MSA
pairwise
distances
time order
tree
redundancy removed by
weights
pseudocounts
or
background-weighted pseudocounts
or
position-specific pseudocounts
account for unobserved data
columns condensed to
profile
adds delete and insert states to
becomes match states in
profile HMM

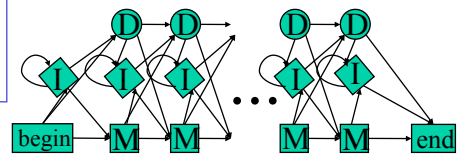
overview

Profile hidden Markov models

I = insertion state

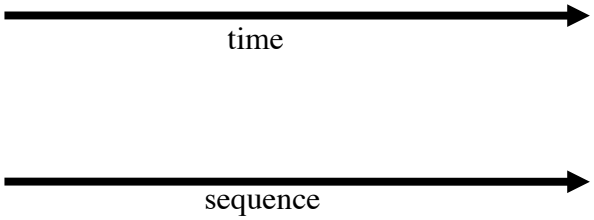
D = deletion state

M = match state



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

Markov processes

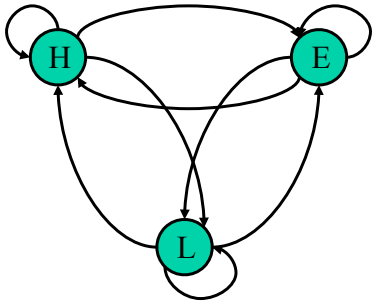


Markov process is any process where the next item in the list depends on the current item. The dimension can be time, sequence position, etc

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

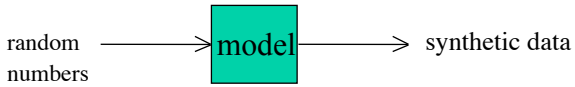
E=extended (strand)

L=loop

What is a stochastic model?

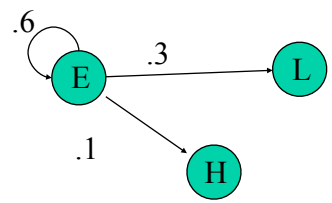
A model is a simplified version of reality. The simpler, the better.

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

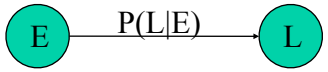
note ===> Markov chains emit **discrete 1-dimensional** data.

Setting the parameters of a Markov model from data.

Secondary structure data

```
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLEEEEELLLLEEEEELLLLLLLLLLLEEEEEEEEELLLLLEEEEEEEELLL
LLLLLLEEEEELLLLLEEEEELLLLLLLLLLLLLLLEEEEELLLLLEEEELLLLLLLLLLEEEELLLLEEEELLLLLLLLLEEEEL
EEEELLLLLLEEEEEEEELLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLLEELLLLLLLLLLLLLEEEEEEEELLLEEEEELLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
HHHHHHHHHHLLLLLLHHHHHHHHHHLLLLLLLLLHHHHHHHHHHHLLLLLHHHHHHHHHHHHLLLLLLLLLLLLLLLLLLLLLL
HHHHHHHHHHLLLLLHHHHHHHHHHHLLLLLLLLLHHHHHHHHHHHLLLLLLHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHLLLLLLHHHHHHHHHHHHHHHHHLLLHHHHHHHHHHHHHHLLLLEEEELLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLHHHHHHHHHHHHHHLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
HHHHHHHHHHHHHHHLLLLLHHHHHHHHHLLLLHHHHHHHLLHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
```

Count the pairs to get the *transition probability*.



$$P(L|E) = P(EL)/P(E) = counts(EL)/counts(E)$$

counts(E) = counts(EE) + counts(EL) + counts(EH)

Therefore: P(E|E) + P(L|E) + P(H|E) = 1.

Bayes' notation and Rabiner's notation

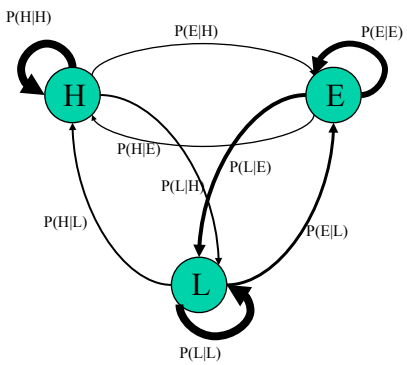
$$a_{yx} = P(x \mid y) = \frac{P(y,x)}{P(y)} = \frac{F(y,x)}{F(y)}$$

...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 depand on

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

λ

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

P("HHEELL"| λ)

=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

P("HHHHHH" | λ) =0.69 ← **common** protein secondary structure

P("HEHEHE" | λ) =1E-6 ← **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

Count the state pairs.

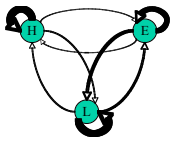
	H	E	L
H	1	1	0
E	0	1	1
L	0	0	1

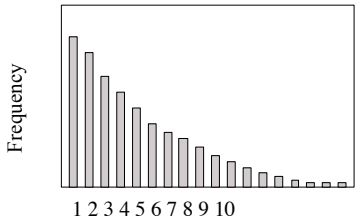
Normalize by row.

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

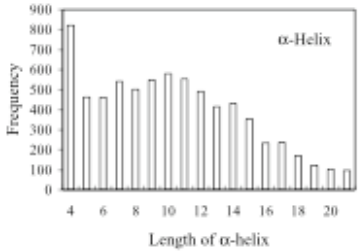
Maximum likelihood model

Is this model too simple?





Synthetic helix length data from this model

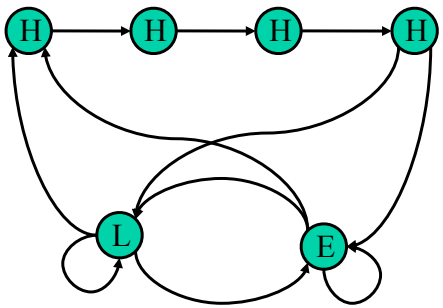


Real helix length data

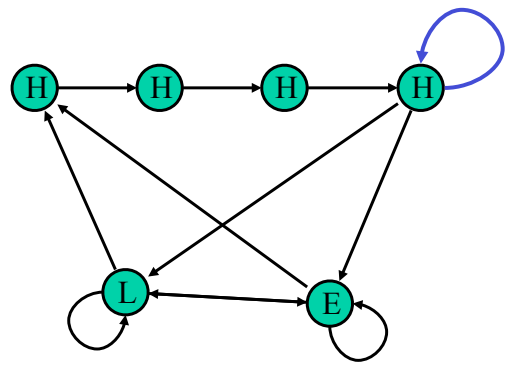
*L.Pal et al, J. Mol. Biol. (2003) 326, 273–291

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

A 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

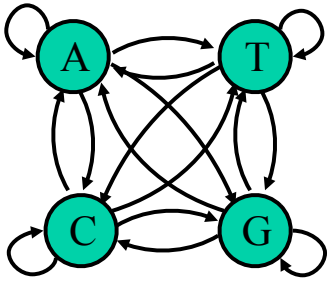


Can you draw a Markov chain where helices are always a multiple of 4 long?

Exercise: generate a MM based on the data.

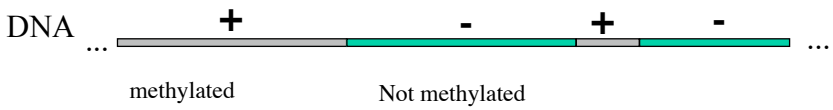
```
how much wood would a wood
chuck chuck if a wood
chuck would chuck wood?
```

Markov chain for DNA sequence



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

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.

NNNCGNNN → NNNTGNNN

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

Using Markov chains for descrimination:

CpG Islands in human chromosome sequences

CpG rich= "+"

+	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	0.385	0.125
T	0.079	0.355	0.384	0.182

CpG poor= "-"

-	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

$$P(CGCG|+) = \pi_C(0.274)(0.339)(0.274) = \pi_C\, 0.0255$$

$$P(CGCG|-) = \pi_C(0.078)(0.246)(0.078) = \pi_C\, 0.0015$$

From Durbin,Eddy, Krogh and Mitcheson “Biological Sequence Analysis” (1998) p.50

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:

β	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.

LLR(CGCG)=**1.812 + 0.461 + 1.812 = 4.085** ← *yes*

In class exercise: what's the LLR?

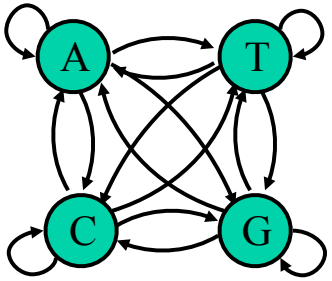
What is the LLR that this seq is a CpG Island?

ATGTCTTAGCGCGATCAGCGAAAGCCACG

β	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**

$$LLR = \sum_{i=1}^{L} \beta_{x_{i-1}x_i} = \_\_\_\_\_$$

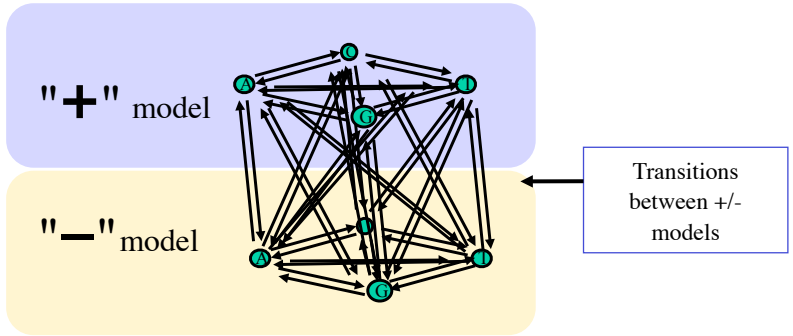
Markov chain for DNA sequence



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

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.... sum these

$$P(CGCG|\lambda) = \sum_{\text{All paths } Q} P(Q)$$

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

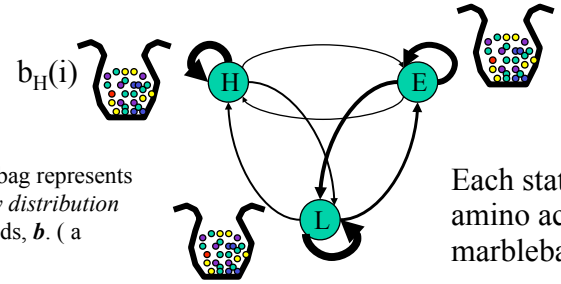
The problem is finding the states given the sequence.

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 <u>impossible</u>, 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?

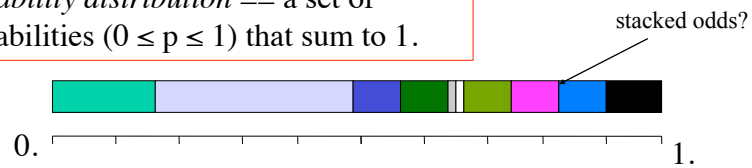
HMM that use profiles



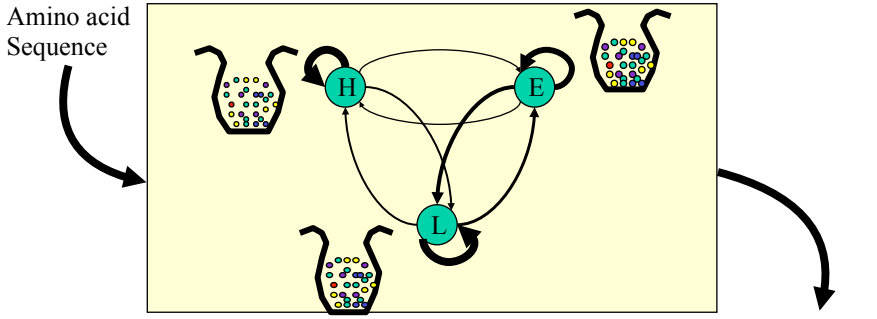
The marble bag represents a *probability distribution* of amino acids, ***b***. (a profile)

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

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



states emit aa and ss.



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

Maximize:

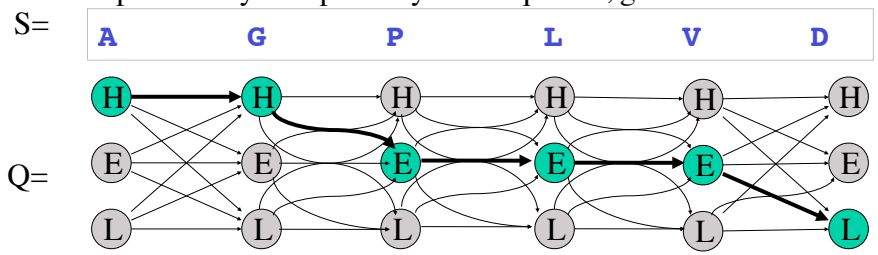
Joint probability of a sequence and pathway

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

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

T = length of S and Q.

Joint probability of a pathway and sequence, given a HMM λ.



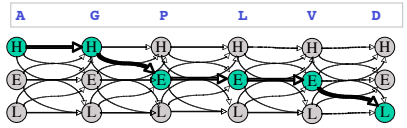
$$P = \pi_H b_H(\mathtt{A}) \times a_{HH} b_H(\mathtt{G}) \times a_{HE} b_E(\mathtt{P}) \times a_{EE} b_E(\mathtt{L}) \times a_{EE} b_E(\mathtt{V}) \times a_{EL} b_L(\mathtt{D})$$

Joint probability : general expression

General expression for pathway Q through HMM λ :

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

**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

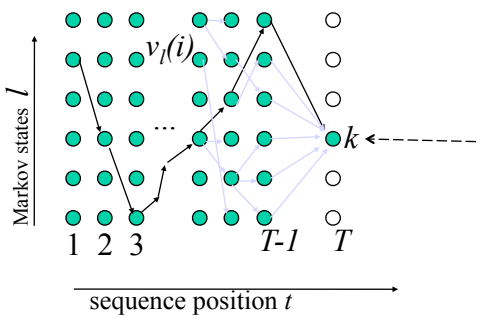
The Viterbi algorithm: the maximum probability path

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

$$Trc_k(t) = \text{ARGMAX}_{l}\ v_l(t\text{-}1)\, a_{lk}\, b_k(s_t)$$

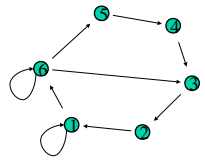
Recursive. We save the value *v* and also a traceback arrow *Trc* as we go along.

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



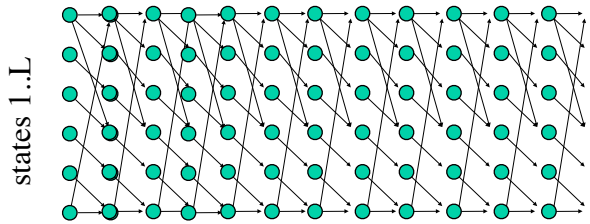
When t = T the last position, the traceback arrow from the MAX give the optimal state sequence.

Exercise: Write the Viterbi algorithm

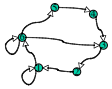


$$v_k(t) = \text{MAX } v_l(t\text{-}1)\, a_{lk}\, b_k(s_t)$$

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

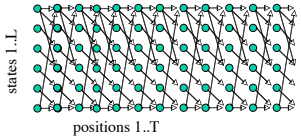


Exercise: Write the Viterbi algorithm



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

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




```
initialize v_k(1)=b_k(s_1)
for t=2,T {
        for k=1,L {

        }
}
```

α

The Forward algorithm:all paths to a state

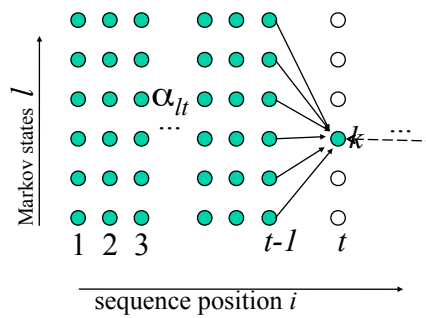
This is *alpha*, the forward probability

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 α depends on the whole previous row of α's.



Sum of P over all paths up to state k at t = $\alpha_k(t)$

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

β

The Backward algorithm:all paths from a state

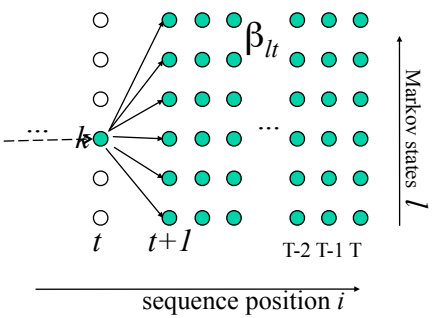
$$\beta_k(t) = \sum_l \beta_l(t+1)\, a_{kl}\, b_k(t)$$

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

Each β depends on the whole ***next*** row of β's.

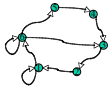
Sum over all paths to state k from t+1

$= \beta_k(t)$

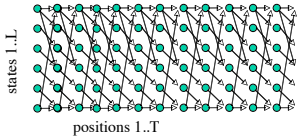


At the beginning of the sequence, when t=1, the sum of $\beta_k(1)$ equals the total probability of the sequence given the model, P(S|λ).

Exercise: Write the Forward algorithm



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



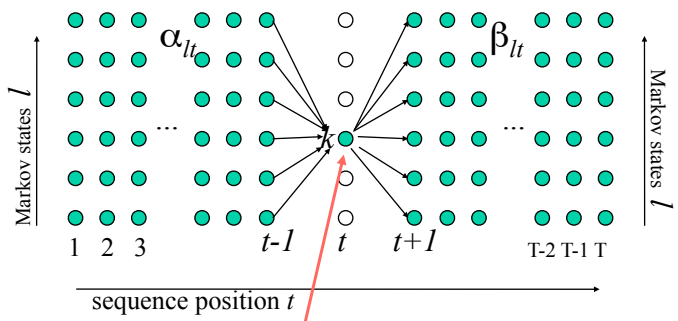

```
initialize α_k(1)=π_k(s_1)

for t=2,T {

        for k=1,L {


        }

}
```

γ

<u>Forward/Backward algorithm</u>: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, λ.



The bottleneck through which all paths must travel.

Expectation/Maximization: refining the model

Example: refining $\boldsymbol{b}_k(G)$ (i.e. the number of Gly's in the kth marble bag)

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

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

Step 3) Do steps 1-2 for all states *k* in λ and all 20 amino acids.

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

Expectation/Maximization is often abbreviated "EM".

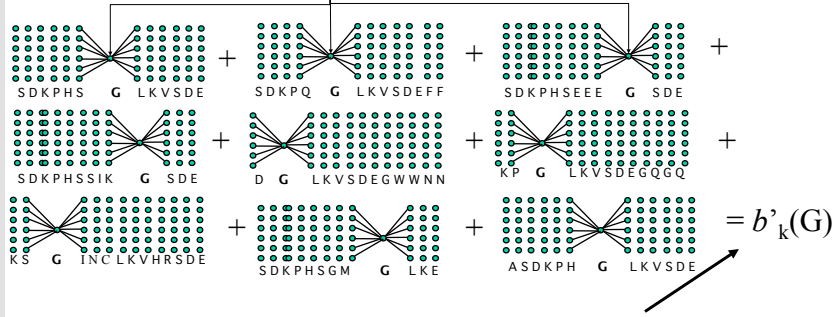
Expectation/Maximization: refining the model

Example: refining $\boldsymbol{b}_k$(G)

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

$P(k|t,S,\lambda) = \Sigma$ all paths though k at $t = \gamma_k(t) = \alpha_k(t) * \beta_k(t)$

Σ over all G in all sequences, S



$= b'_k(G)$

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

Expectation/Maximization: refining the model

Example: refining $\boldsymbol{a}_{jk}$, 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 $\boldsymbol{a}_{jk}$

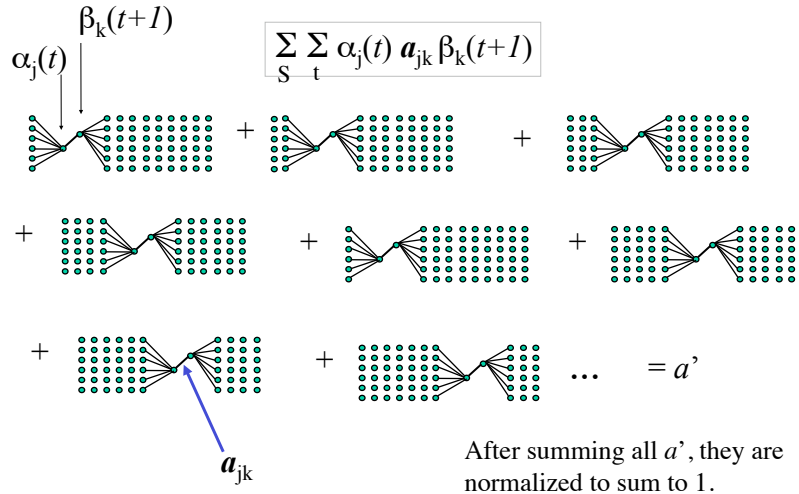
Step 4) Do Steps 1-3 for all positions t and all sequences, S.
Sum--> $\boldsymbol{a}$'. Then normalize. Reset $\boldsymbol{a}_{jk}$ in the new model to $\boldsymbol{a}$'.

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

Expectation/Maximization: refining the model

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

$$\sum_S \sum_t \alpha_j(t)\, \boldsymbol{a}_{jk}\, \beta_k(t+1)$$



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

Σ over all t in all sequences, S

“Profile HMMs”

State emissions:

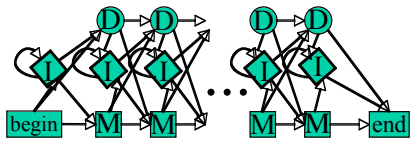
I = insert state, one character from the background profile

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

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

Begin = non-emitting. Source state.

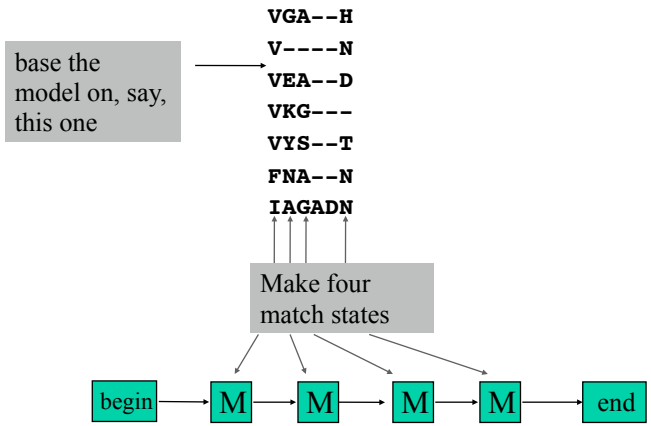
End = non-emitting. Sink state.



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(End)*. This is the measure of how well the <u>sequence</u> fits the <u>model</u>. Then we can test several models.

Generating a profile HMM from a multiple sequence alignment



Generating a profile HMM from a multiple sequence alignment

base the model on, say, this one

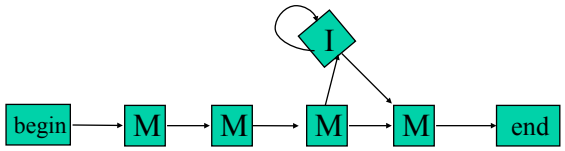
```
VGA--H
V----N
VEA--D
VKG---
VYS--T
FNA--N
IAGADN
```

Make four match states

Generating a profile HMM from a multiple sequence alignment

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

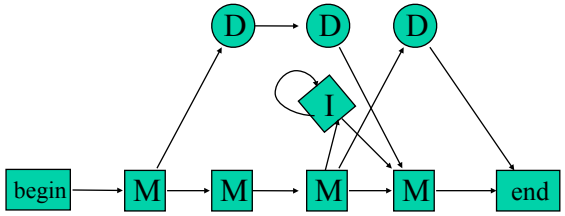
Add insertion states where there are insertions. (red)



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)

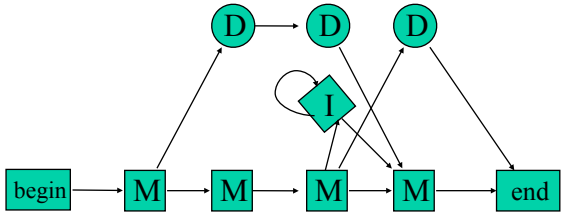


...now optimize using *expectation maximization.*

Getting profiles for every Match state

w_1	VGA--H
w_2	V----N
w_3	VEA--D
w_4	VKG---
w_5	VYS--T
w_6	FNA--N
w_7	IAGADN

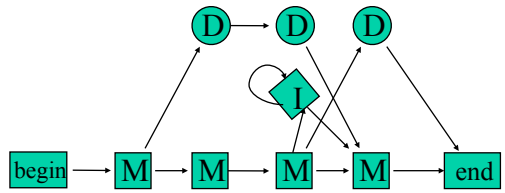
Count the frequency of each amino acid, scaled by sequence weights, *w*.



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

$\boldsymbol{b}_{M1}(V) = (w_1+w_2+w_3+w_4+w_5)/$
$(w_1+w_2+w_3+w_4+w_5+w_6+w_7)$

Calculating the probability of a sequence given the model: P(s|λ)



<u>Sum forward (forward algorithm) using the sequence *s*.</u>

For each <u>M</u>atch state, multiply by the transition (*a*) and the profile value, $\boldsymbol{b}_M(s_i)$, and increment *i*

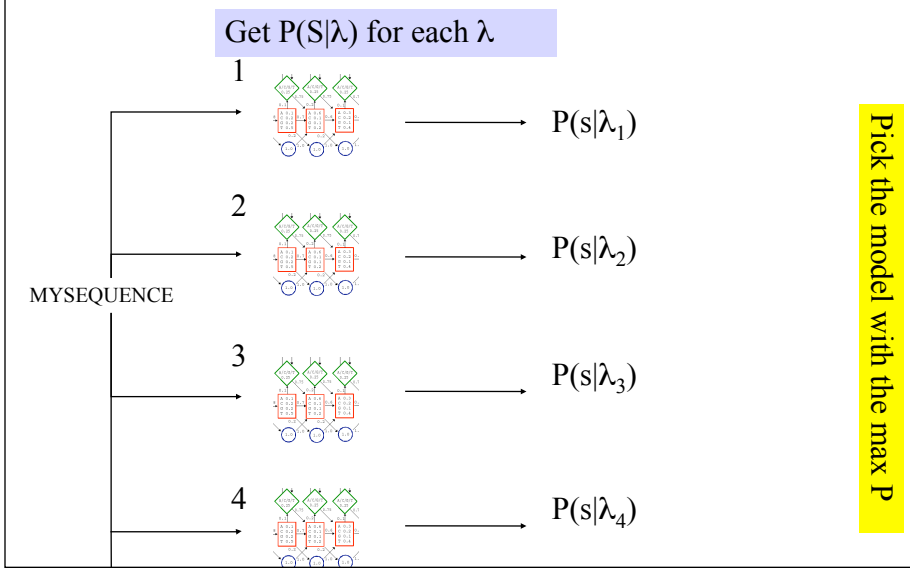
For each <u>D</u>eletion state, multiply by *a*, do not increment *i*.

For each <u>I</u>nsertion 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



In Class exercise: make a profile HMM

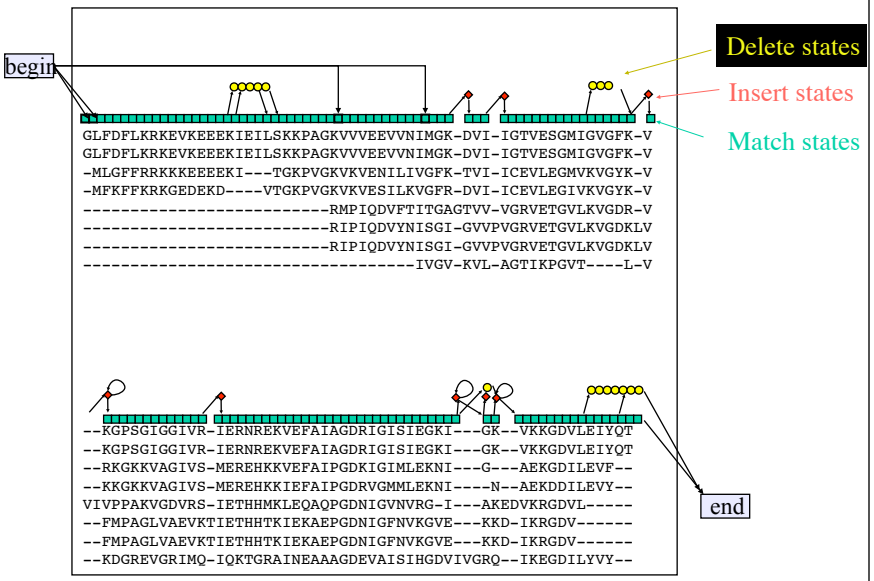
```
AGF---PDG
AGGYL-PDG
AG----PNG
SGFFLIPNG
SGF--EPNG
```

- 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   GLFDFLKRKEVKEEEKIEILSKKPAGKVVVEEVVNIMGK-DVI-IGTVESGMIGVGFK-V 59
18977279 2   GLFDFLKRKEVKEEEKIEILSKKPAGKVVVEEVVNIMGK-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-MEREHKKIEFAIPGDRVGMMLEKNI----N--AEKDDILEVY-- 99
18313751 275 VIVPPAKVGDVRS-IETHHMKLEQAQPGDNIGVNVRG-I---AKEDVKRGDVL----- 322
729396   269 --FMPAGLVAEVKTIETHHTKIEKAEPGDNIGFNVKGVE---KKD-IKRGDV------ 314
1361925  272 --FMPAGLVAEVKTIETHHTKIEKAEPGDNIGFNVKGVE---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

SAM:

www.cse.ucsc.edu/research/compbio/HMM-apps/