Bioinformatics 1 -- Lecture 18

more HMMs...
Dealing with underflow
Pfam.
MARCOIL, TMHMM
Bayes Block Alignment.

<u>Unrolling the Viterbi algorithm</u>: underflow problems solved by going to log space

Algorithm:

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

Algorithm unrolled:

$$v_k(2) = b_{q_1}(s_1)\, a_{q_1 k}\, b_k(s_2)$$

$$v_k(3) = b_{q_1}(s_1)\, a_{q_1 q_2}\, b_{q_2}(s_2)\, a_{q_2 q_3}\, b_{q_3}(s_3)$$

$$v_k(4) = b_{q_1}(s_1)\, a_{q_1 q_2}\, b_{q_2}(s_2)\, a_{q_2 q_3}\, b_{q_3}(s_3)\, a_{q_3 q_4}\, b_{q_4}(s_4)$$

...

Why will this calculation fail for $v_k(100)$?
Hint: Try multiplying 200 numbers (all between 0 and 1) together.

Viterbi algorithm underflow: log space solution

Algorithm:

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

Log space:

$$Log(v_k(t)) = \max_l [\, \mathrm{Log}(v_l(t-1)) + \mathrm{Log}(a_{lk}) + \mathrm{Log}(\, b_k(s_t)) \,]$$

<u>The Forward algorithm</u>: underflow problem

Algorithm:

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

Algorithm unrolled:

$$\alpha_k(2) = b_k(2)\, [\alpha_1(1)\, a_{1k} + \alpha_2(1)\, a_{2k} + \alpha_3(1)\, a_{3k} + \alpha_4(1)\, a_{4k}\,]$$

$$\begin{aligned}\alpha_k(3) = b_k(3)\, [\, & b_1(2)\, [\alpha_1(1)\, a_{11} + \alpha_2(1)\, a_{21} + \alpha_3(1)\, a_{31} + \alpha_4(1)\, a_{41}\,]\, a_{1k} + \\ & b_2(2)\, [\alpha_1(1)\, a_{12} + \alpha_2(1)\, a_{22} + \alpha_3(1)\, a_{32} + \alpha_4(1)\, a_{42}\,]\, a_{2k} + \\ & b_3(2)\, [\alpha_1(1)\, a_{13} + \alpha_2(1)\, a_{23} + \alpha_3(1)\, a_{33} + \alpha_4(1)\, a_{43}\,]\, a_{3k} + \\ & b_4(2)\, [\alpha_1(1)\, a_{14} + \alpha_2(1)\, a_{24} + \alpha_3(1)\, a_{34} + \alpha_4(1)\, a_{44}\,]\, a_{4k}\ \,]\end{aligned}$$

…

Can't do this one in Log space,
because Log(a+b) can't be simplified.

Solution: <u>dynamic scaling</u> to keep numbers within a reasonable range

The Forward algorithm: underflow problem, scaling solution

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

Let, $\alpha'_k(2) = c_2 \sum \alpha_l(1)\, a_{lk}\, b_k(2)$

where c_1 *is any scale factor.*

And, $\alpha'_k(3) = c_3 \sum \alpha'_l(2)\, a_{lk}\, b_k(3)$

where c_2 *is any scale factor.*

And so on, for all sequence positions t.

Then, $\alpha'_k(t) = c_t \sum \alpha'_l(t-1)\, a_{lk}\, b_k(t-1)$

If we choose, $c_t = 1/\sum \alpha'_k(t)$,

then, $c_t\, \alpha'_k(t)$ has a mean value of 1.

<u>The Forward / Backward algorithm</u>: scaling solution does not change gamma.

Re-writing, $\alpha'_k(t) = c_t \sum_l (\Pi_{i=1,t-1} c_i)\, \alpha_l(t-1)\, a_{lk}\, b_k(t-1)$

So, $\alpha'_k(t) = (\Pi_{i=1,t}\, c_i)\, \alpha_k(t)$

If we apply the same scale factors to the backward value β, then, $\beta'_k(t) = (\Pi_{i=t,T}\, c_i)\, \beta_k(t)$

Then the calculation of the *a posteriori* value γ, is

$\gamma'_k(t) = \alpha'_k(t)\, \beta'_k(t) = (\Pi_{i=1,t}\, c_i)\, \alpha_k(t) (\Pi_{i=t,T}\, c_i)\, \beta_k(t)$

$= (\Pi_{i=1,t}\, c_i)\, (\Pi_{i=t,T}\, c_i)\, \alpha_k(t)\, \beta_k(t)$

$= (\Pi_{i=1,T}\, c_i)\; c_t\, \alpha_k(t)\, \beta_k(t) = C_T\, c_t\, \alpha_k(t)\, \beta_k(t) = C_T\, c_t\; \gamma_k(t)$

where, $C_T = (\Pi_{i=1,T}\, c_i)$

Since γ is normalized to sum to 1,

$\gamma'_k(t) = C_T\, c_t\; \gamma_k(t) / \Sigma_k\, C_T\, c_t\; \gamma_k(t) = \gamma_k(t)$

Gamma from scaled summation is the same as gamma unscaled.

Example HMM data structure

in fortran...

```
type HMMSTATE
     integer :: id, ntrans
     type (HMMSTATE), pointer :: q(:)
     real :: a(:), b(20), emit(:)
     logical :: emitting
end type HMMSTATE
type (HMMSTATE):: hmm_root
```

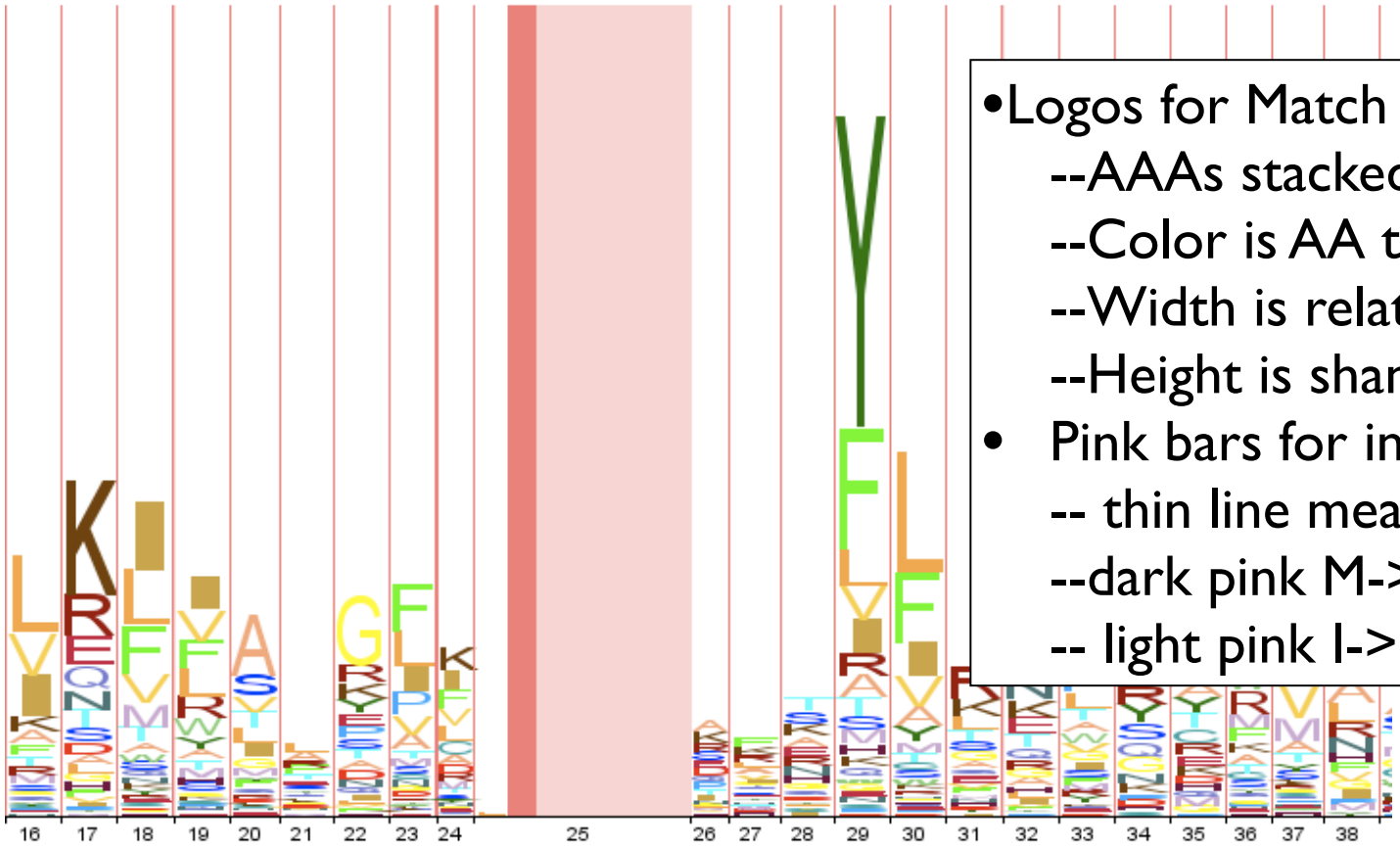
Explanation:
This is a multi-linked list. *hmm_root* should be the "begin" state, with *hmm_root %emitting*=.false. There are *ntrans* transitions probabilities *a*(:). *q*(:) points to the corresponding next state. If (emitting==.true.) then the state emits amino acid profile *b*, and optionally something else called *emit*(:).

Pfam: Protein families

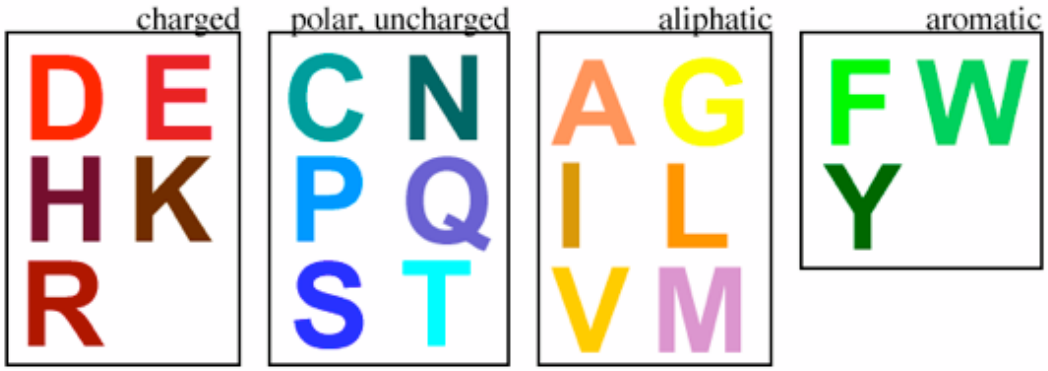
A searchable database of multiple sequence alignments and profile-HMMs.

database	sequence alignments from	curated?
Pfam-A	UniProtKB	yes
Pfam-B	ADDA (Holm)	no

PFAM visualization of profile HMM

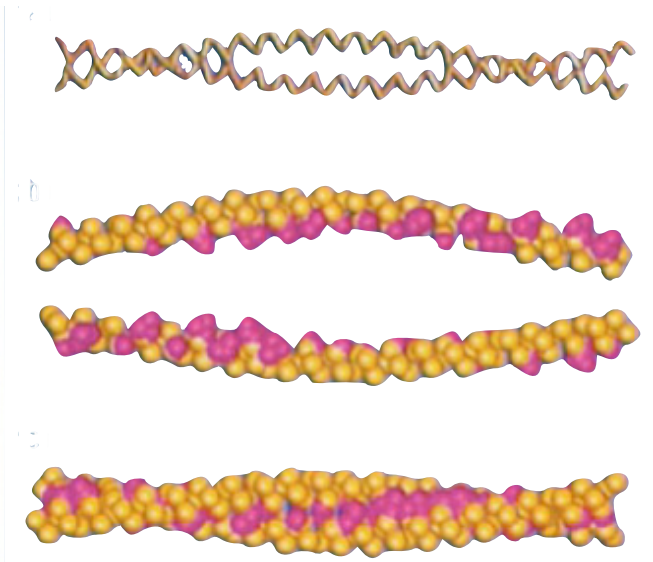


- Logos for Match states.
  - --AAAs stacked by probability
  - --Color is AA type
  - --Width is relative contribution.
  - --Height is shannon entropy
- Pink bars for insert states
  - -- thin line means no I state
  - --dark pink M->I,
  - -- light pink I->I



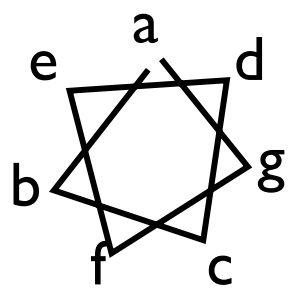
MARCOIL predicts coiled coils

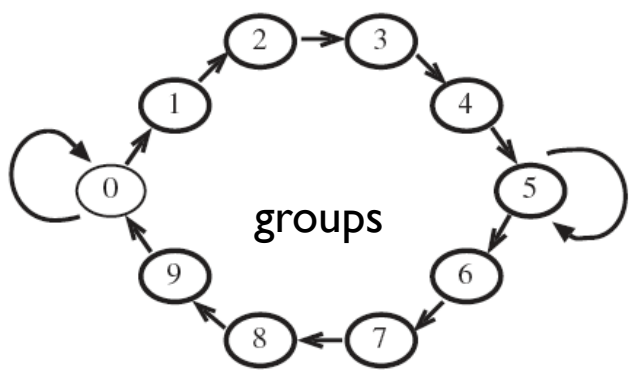
MARCOIL consists of 9 groups of 7 states. Each of the 7 models a position in the helix, a-f. There are 4 special pre-coil and 4 post-coil groups, one repeating coil state (5), and one generic state (0).



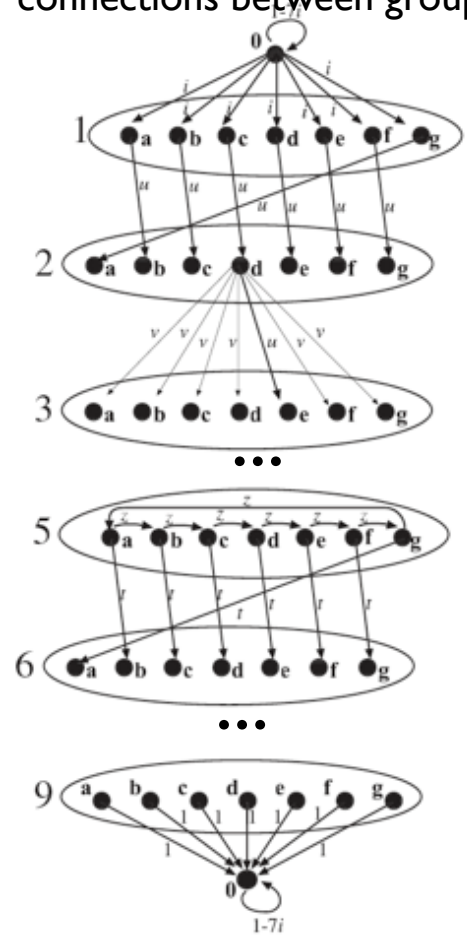
From **Protein Structure and Function**
by Gregory A Petsko and Dagmar Ringe

Helix positions





connections between groups



MARCOIL. Delorenzi & Speed, 2002

HMM for dicodon preferences

- Codon preferences exist due to differences in [tRNA] in the cell.
- Di-codon preferences exist due to interactions between neighboring tRNAs on the ribosome.
- Di-codon preferences are preserved* in the DNA sequences of ORFs in the genome.
- To find the optimal set of codons for a protein sequence, design a HMM based on codons, emitting amino acids in parallel. A parallel HMM. Maximum likelihood transitions between codons are the dicodon frequencies.
- Use Viterbi to assign codons to an amino acid sequence.

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

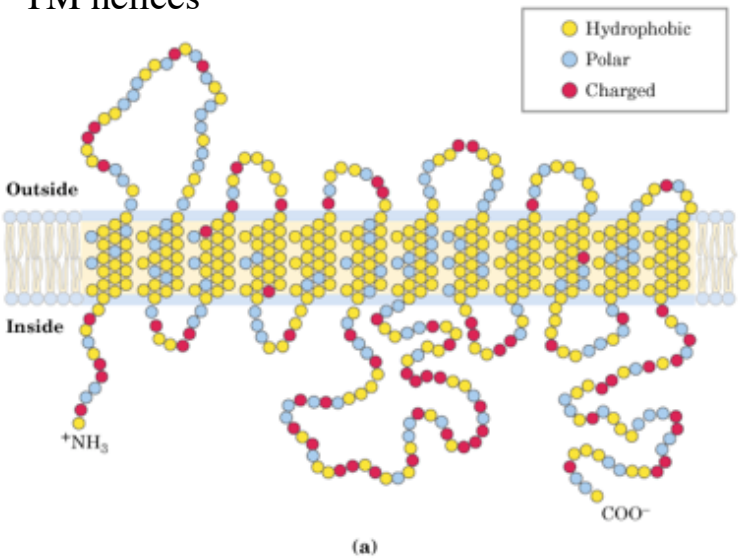
l= all codons

dicodon frequencies

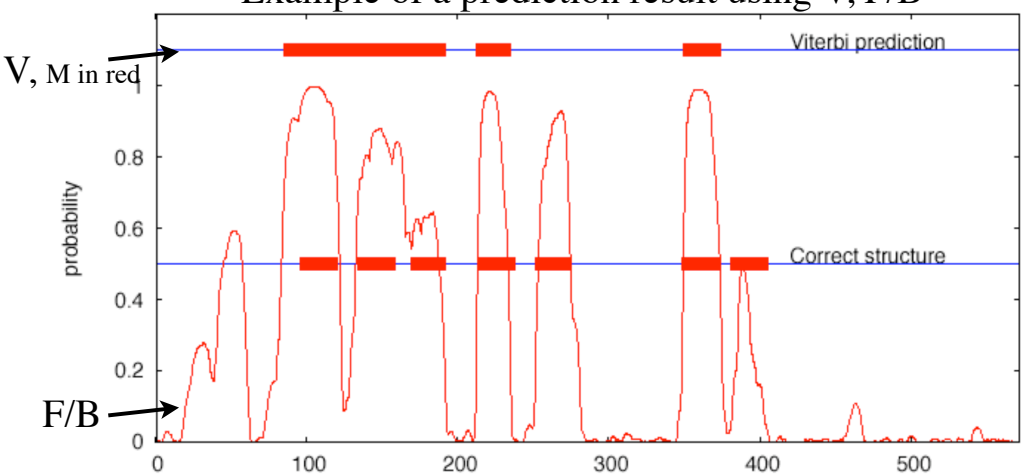
emissions = 1.00 or 0.00

TMHMM -- transmembrane helices

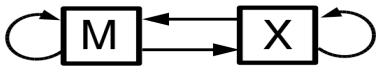
TM helices



Example of a prediction result using V, F/B



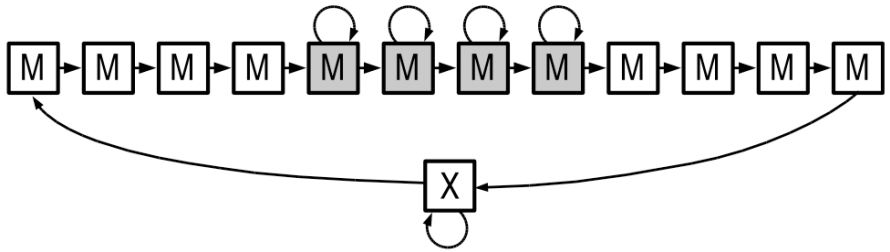
TMHMM (Krogh, 2001) models transmembrane helices in eukaryotic and inner-bacterial membranes. First attempt was a composition-based model, 2 states.



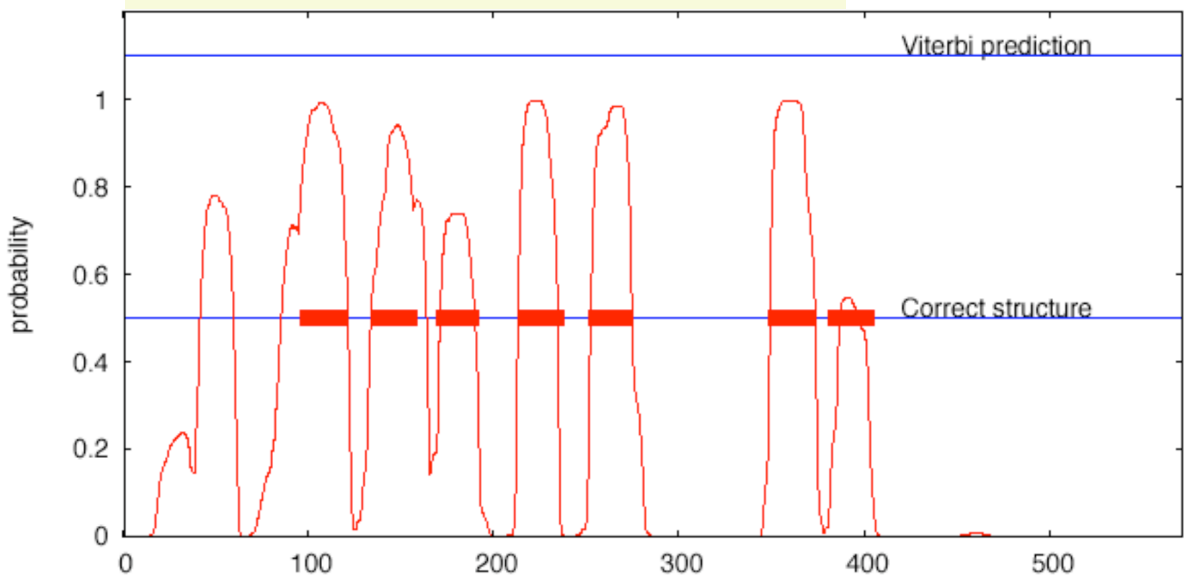
	TM helices		Other regions		Over-rep
AA	count	freq	count	freq	
I	1826	0.120	2187	0.046	2.61
F	1370	0.090	1854	0.039	2.31
L	2562	0.168	4156	0.087	1.93
V	1751	0.115	2935	0.061	1.89
M	616	0.040	1201	0.025	1.60
W	414	0.027	819	0.017	1.59
A	1657	0.109	3382	0.071	1.54
Y	615	0.040	1616	0.034	1.18
G	1243	0.082	3352	0.070	1.17
C	289	0.019	960	0.020	0.95
T	755	0.050	2852	0.060	0.83
S	806	0.053	3410	0.071	0.75
P	423	0.028	2640	0.055	0.51
H	121	0.008	1085	0.023	0.35
N	250	0.016	2279	0.048	0.33
Q	141	0.009	2054	0.043	0.21
D	104	0.007	2551	0.053	0.13
E	110	0.007	2983	0.062	0.11
K	78	0.005	2651	0.055	0.09
R	83	0.005	2933	0.061	0.08
Tot	15214	1.000	47900	1.000	

TMHMM -- 15 state version

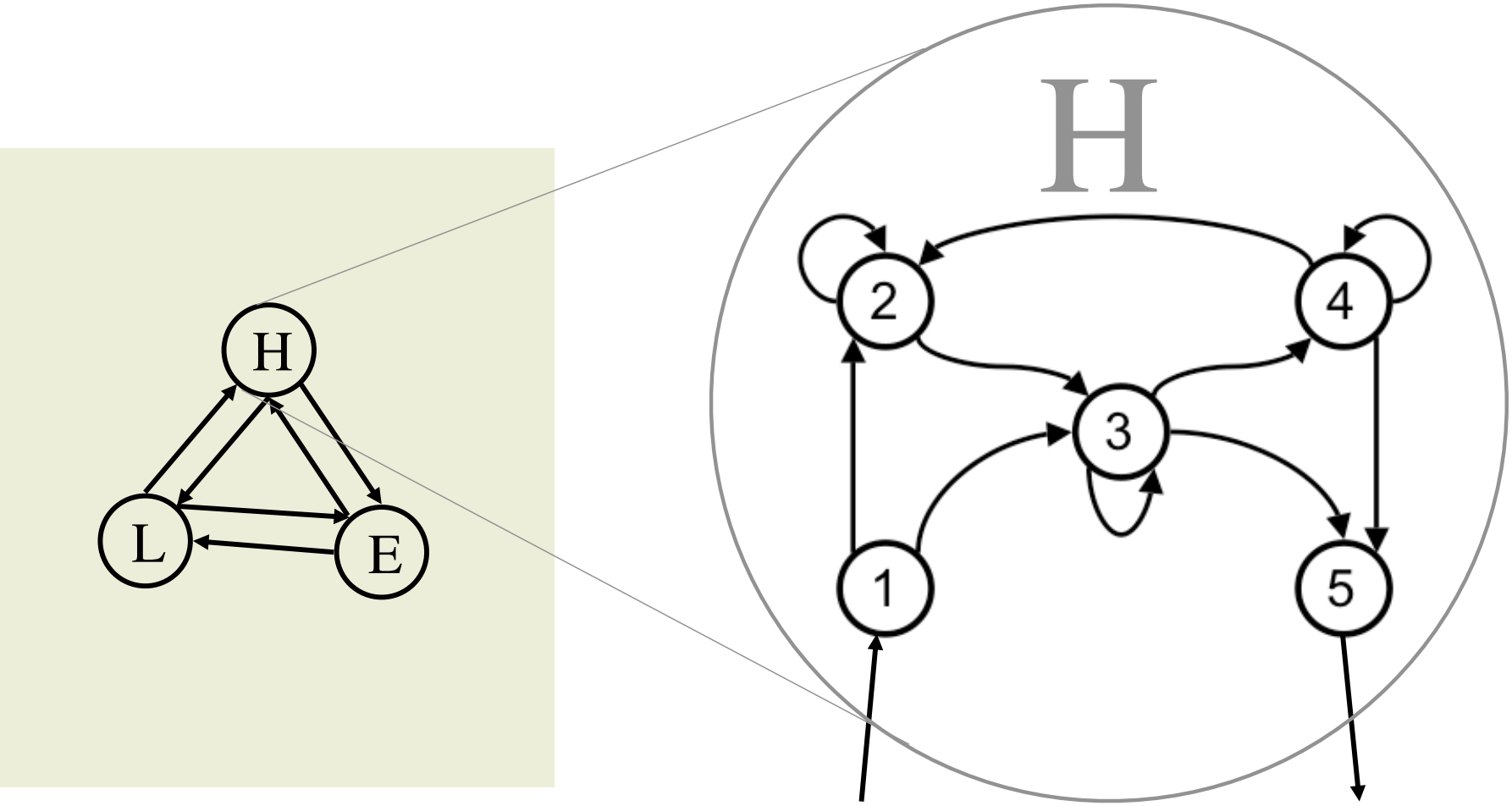
New version has specific pre-helix and post-helix states and a more reasonable M-length distribution, ranging from 14-28+, not 1-28+



Better F/B prediction, but V is worse. Why?



HMMs within HMMs



We can define a very simple HMM for secondary structure. But here, each state is a macro-state emitting a variable length string of amino acids, from an internal HMM.

The topology of the helix (H) unit used by **Asai et al** [1993] to predict secondary structure. The periodicity of amphipathic helices is approximately modeled by the cycle of states. States 1 and 5 represent the start and end of the helix, respectively.

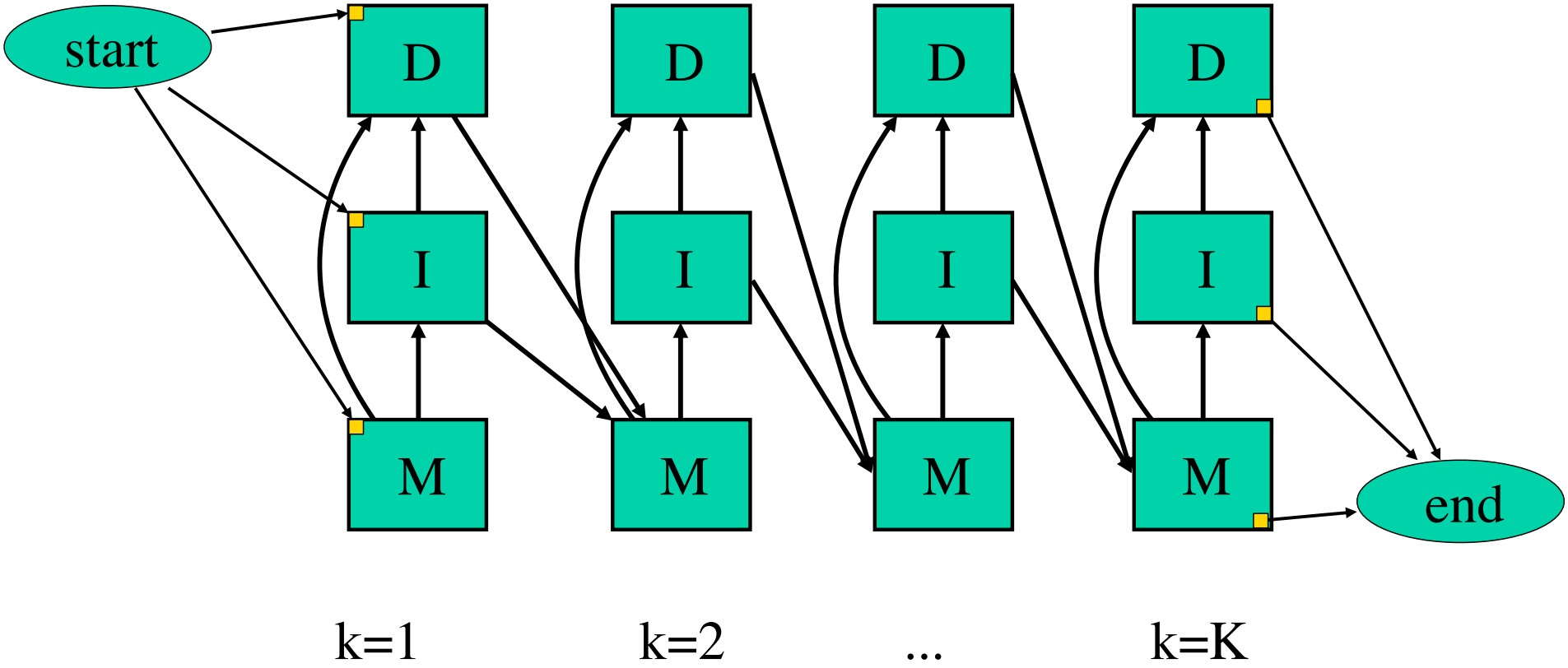
Other topologies were explored for E and L macro-states.

Suboptimal alignments using Bayes Block Alignment.

Bayes Block Alignment (BBA)

Find all alignments that have at most K gaps.

Sankoff, 1972; Zhu, Liu & Lawrence, 1998



Maximum number of indels = K = 20 or L/10, whichever is less.

Algorithm for BBA <u>forward probabilities</u>

D

$$D[k,i,j] = \sum \begin{cases} M[k,i-1,j] \\ I[k,i-1,j] \\ D[k,i-1,j] \end{cases}$$

I

$$I[k,i,j] = \sum \begin{cases} M[k,i,j-1] \\ I[k,i,j-1] \end{cases}$$

M

$$M[k,i,j] = LR(i,j) \times \sum \begin{cases} M[k,i-1,j-1] \\ I[k-1,i-1,j-1] \\ D[k-1,i-1,j-1] \end{cases}$$

LR(i,j) = likelihood ratio = substitution probability

The forward product must be scaled

Underflow problems....

Solution: re-scaling at each step and saving the scale factors.

See slide 5

Algorithm for BBA forward sums: M

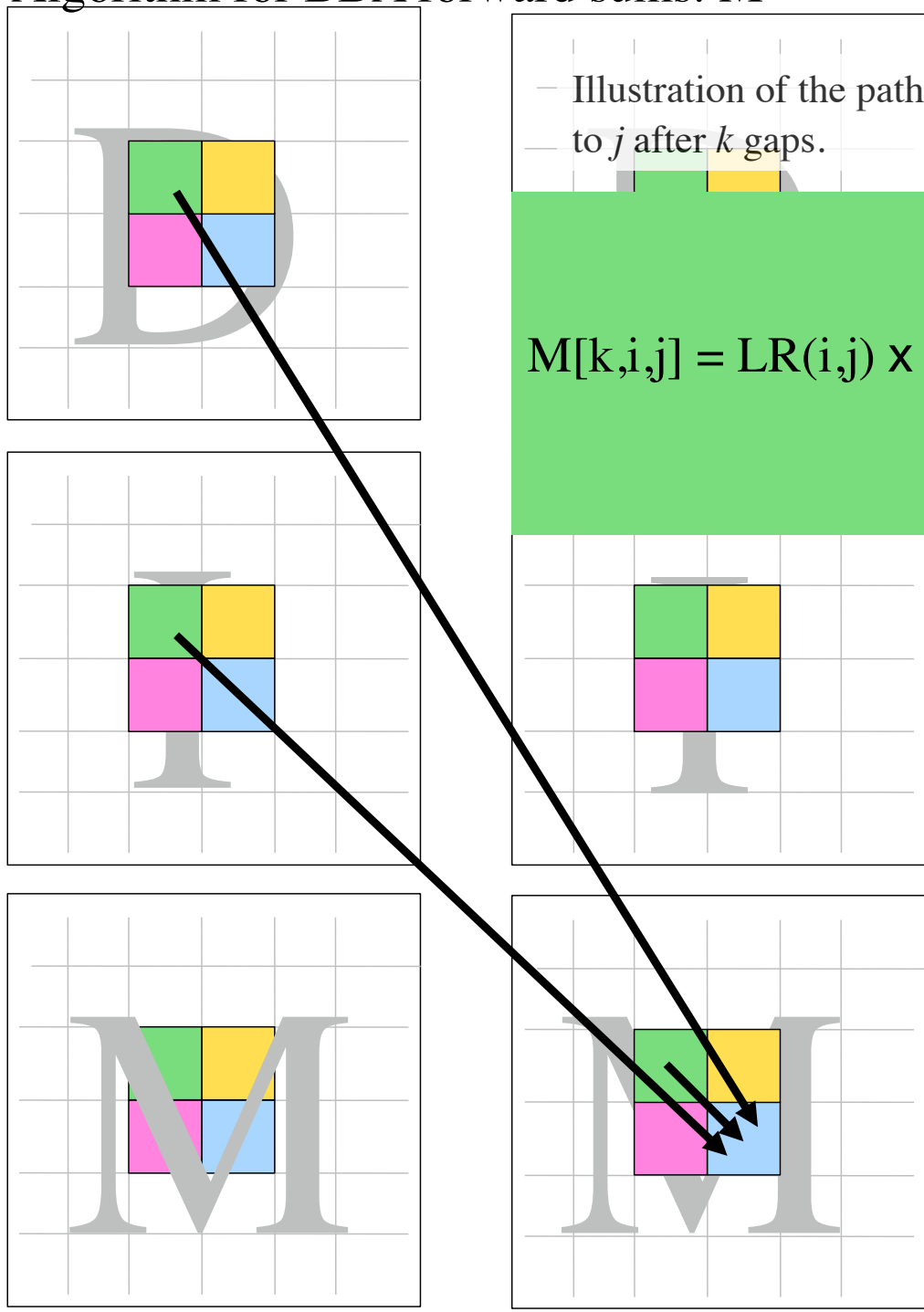


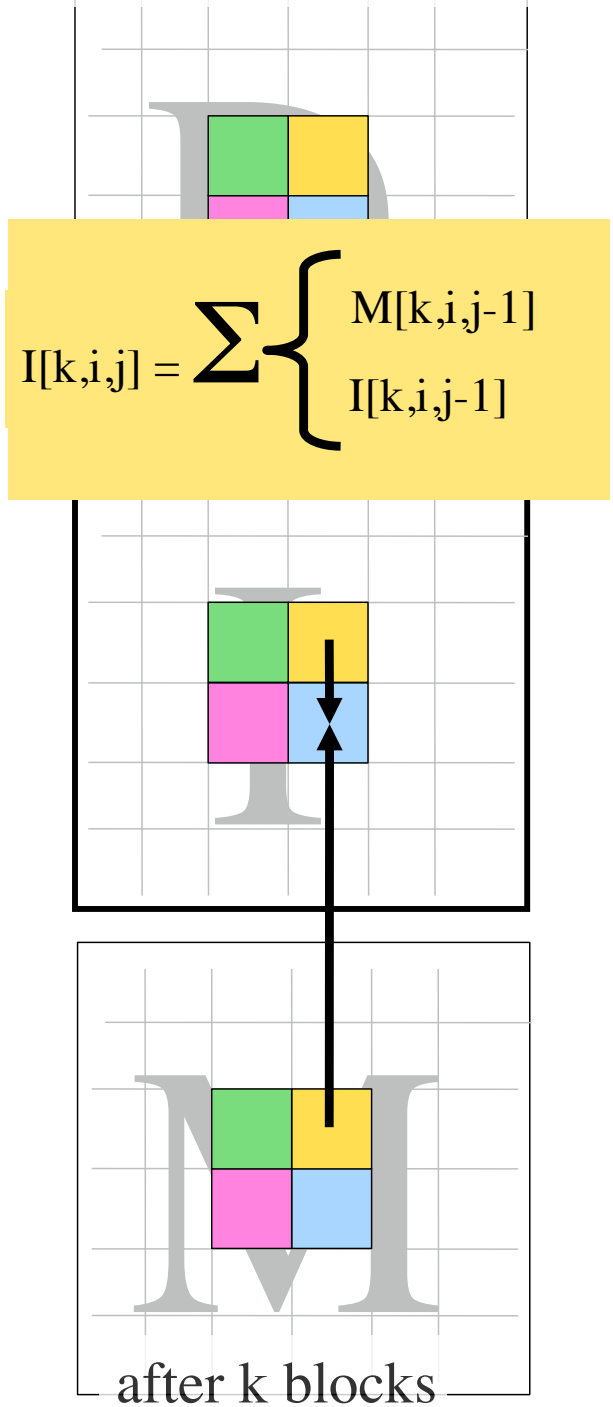
Illustration of the paths into M[k,i,j], alignment of *i* to *j* after *k* gaps.

$$M[k,i,j] = LR(i,j) \times \sum \begin{cases} M[k,i-1,j-1] \\ I[k-1,i-1,j-1] \\ D[k-1,i-1,j-1] \end{cases}$$

indel after k blocks

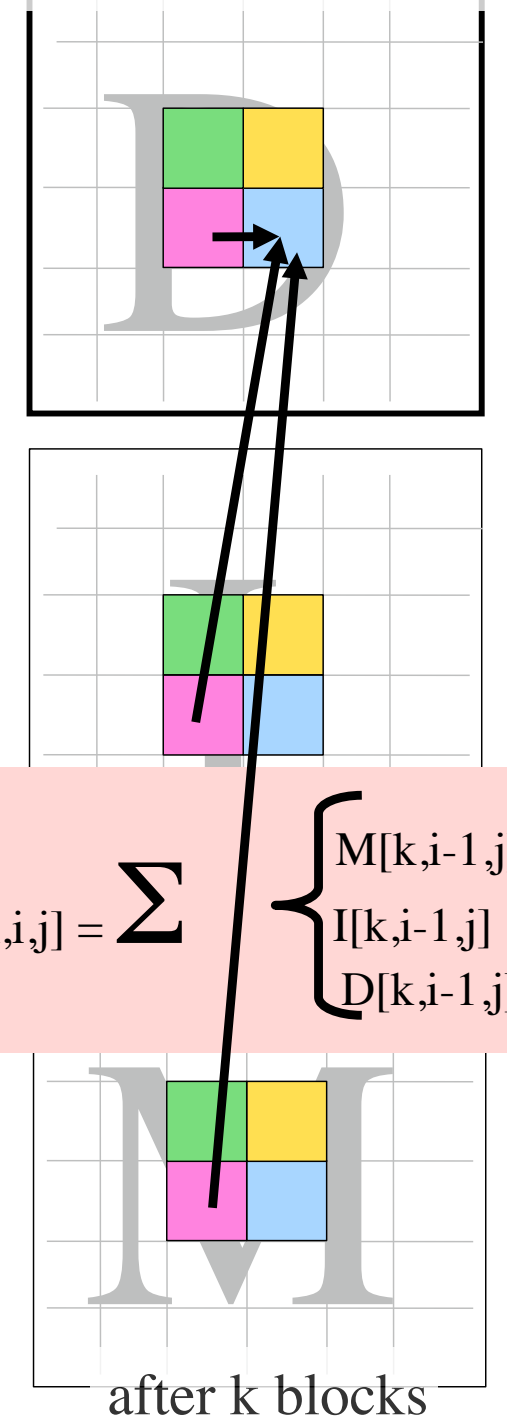
match after k gaps

Algorithm for BBA forward sums: I



$$I[k,i,j] = \sum \begin{cases} M[k,i,j-1] \\ I[k,i,j-1] \end{cases}$$

Algorithm for BBA forward sums: D



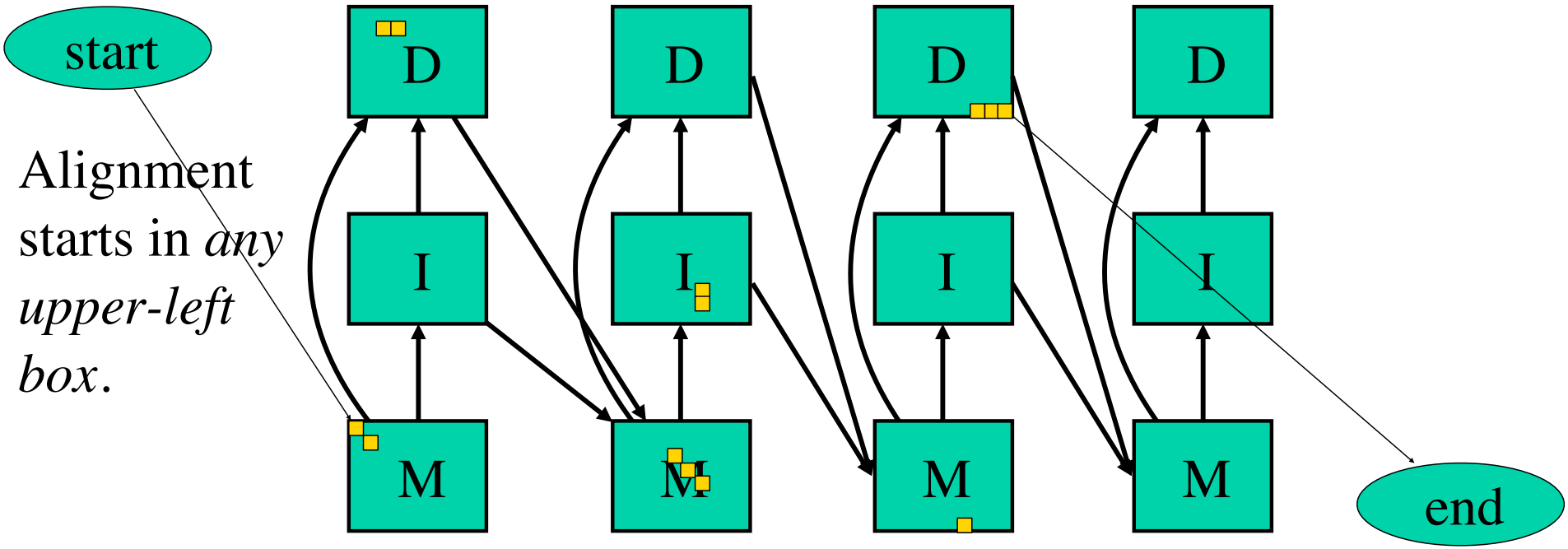
$$D[k,i,j] = \sum \begin{cases} M[k,i-1,j] \\ I[k,i-1,j] \\ D[k,i-1,j] \end{cases}$$

"**sampleback**" instead of traceback.

```
Start "sampleback" at lower right, M(K,I,J),D(K,I,J), or I(K,I,J)
then do the following:
i=I;  j=J; k=K; histogram(..)=0;
While (i>0 and j>0) do
 If current state is M,
  add 1 to histogram(i,j)
  y=D(k-1,i-1,j-1)+I(k-1,i-1,j-1)+M(k,i-1,j-1)
  x = random number 0≤x≤1.
  If (x < D(k-1,i-1,j-1)/y) then
    next state is D(k-1,i-1,j-1)
  else if (x<(D(k-1,i-1,j-1)+ I(k-1,i-1,j-1))/y) then
    next state is I(k-1,i-1,j-1)
  else
    next state is M(k,i-1,j-1)
  end if
 else if current state is I, ... (try filling this in)
 else if current state is D, ... (try filling this in)
end do
```

Illustration of one sample-back

do this 10,000 times



Alignment starts in *any upper-left box*.

small orange boxes show a path through the blocks

```
MMDDMMMIIMDDD
AGCGCGC~~TTCA
AG~~CGCCCT~~~
```

Alignment ends in *any lower-right box*.

Result is a histogram , P(i,j)

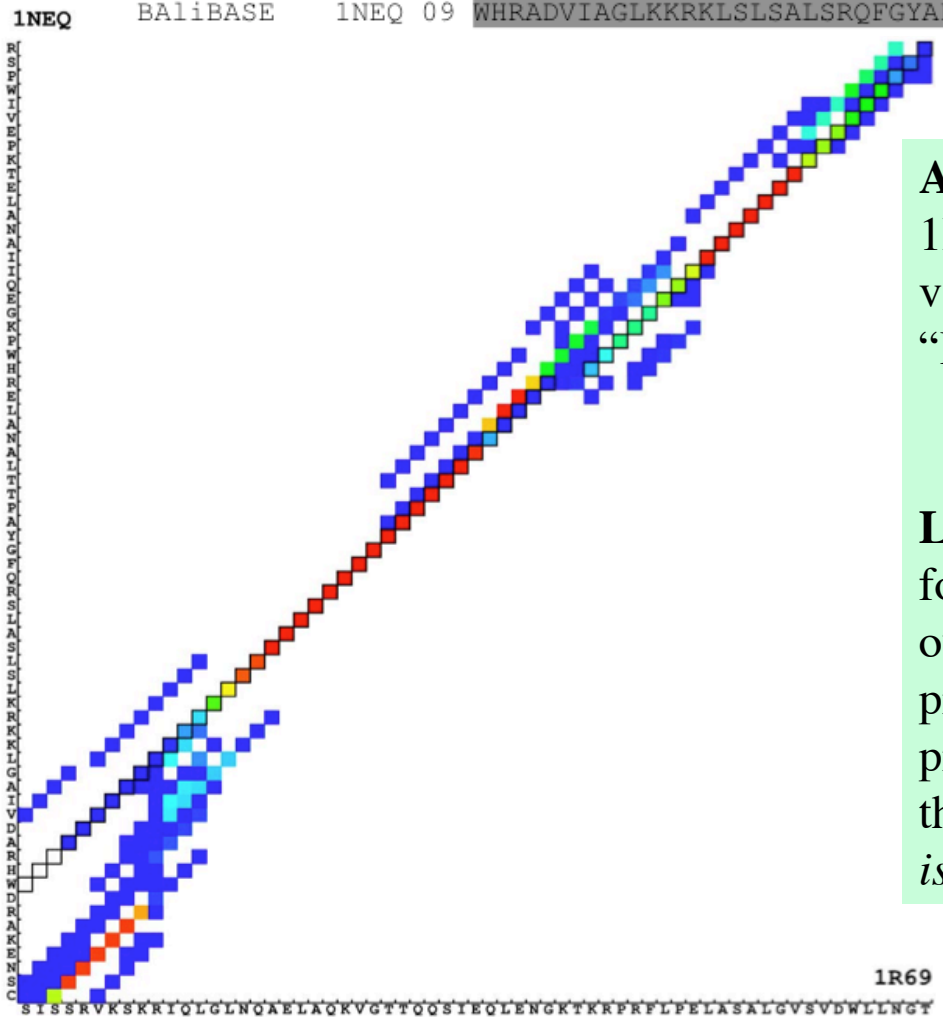
We can plot the probability of a match for every *ij*. In the example below *the proteins are NOT homologous*, but short stretches of similarity are found nonetheless.

http://www.bioinfo.rpi.edu/applications/bayesian/bayes/manual/phylogenetic_help.html

Testing DP versus BBA for hard cases

```
                1R69 01 SISSRVKSKRIQLGLNQAELAQKVGTTQQSIE-Q-LENGKTKRPRFLPELASALGVSVDWLLNGT
BLOSUM40        1NEQ 17 ---------------------------------------GLKKRKLSLSALSRQFGYAPTTLANA-
SDM             1NEQ 31 -----------QFGYAPTTLANALERHWPKGE-QIIANALETKPEVI------------------
HMMSUM-D3       1NEQ 13 DVIAGLKKRKLSL----SALSRQFGYAPTTLANA-LERHWPKGEQII---ANALETKPEVIWPSR
HMMSUM-D3+NS    1NEQ 13 DVIAGLKKRKLSLSALSRQFGYAPTTLANALE-R-------HWPKGEQIIANALETKPEVIWPSR
HMMSUM-D        1NEQ 14 -----VIAGLKKRKLSLSALSRQFGYAPTTLA-N-ALERHWPKGEQIIANALETKPEVIWPSR--
HMMSUM-DNS      1NEQ 11 --RADVIAGLKKRKLSLSALSRQFGYAPTTLA-N-ALERHWPKGEQI--IANALETKPEVIWPSR
BAliBASE        1NEQ 09 WHRADVIAGLKKRKLSLSALSRQFGYAPTTLA-N-ALER--HWPKGEQIIANALETKPEVIWPSR
```

(Huang & Bystroff, 2006)



Above: Optimal DP alignments of 1R69 to 1NEQ (distant homolog proteins) using various subsitution matrices. The last line "BAliBase" is the true alignment.

Left: Bayesian Adaptive Alignment results for the same pair. True alignment in black outline. Color indicates (white: zero probability, blue: low P, red: high P) probability of a match between positions in the sequences. *Sometimes the true alignment is sub-optimal*.

BBA is better than DP.