Plan

- Problem of sequence alignment
  - Algorithm
  - Global Alignment
  - Local Alignment
- Substitution matrices
- Fast database search: BLAST

Sequence Alignment

The Problem: Given:

$\mathbf{a}$ = `MVPAGIW`

$\mathbf{b}$ = `MVAGLRW`

find *the best* alignment:

$\mathbf{a}^*$ = `MVPAGI-W`

$\mathbf{b}^*$ = `MV-AGLRW`

6 matches

5 identities

1 substitution (I ↔ L)

2 gaps (P ↔ - and R ↔ -)

Scoring

$S =$ #identities $+\mu$ #substitutions $-\delta$ #gaps

Number of possible alignments

$$\Omega = \binom{M+N}{N} = \frac{(M+N)!}{M!N!}$$

$$\Omega(M = N = 100) \approx 10^{59}$$

BUT: Dynamic programming can find
the optimal solution!

(Due to local additivity of the scoring function and the lack of “loops”).

Global Alignment Needlman-Wunsch

Given:

Sequences: $\mathbf{a} = a_1 a_2 \cdots a_n$; $\mathbf{b} = b_1 b_2 \cdots b_m$.

Matrix: $s(x, y)$ and gap penalty $s(x, -) = s(-, x) = g(x)$

Find:

$$S(\mathbf{a}, \mathbf{b}) = max \sum_{i=1}^{L} s(a_i^*, b_i^*)$$

Solution:

1. Define $S_{ij} = S(a_1 a_2 \cdots a_n, b_1 b_2 \cdots b_m)$
and set $S_{00} = 0$,
$S_{0j} = \sum_{k=1}^{j} s(-, b_k)$, and $S_{i0} = \sum_{k=1}^{i} s(a_k, -)$
2.

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j), \\ S_{i-1,j} + s(a_i, -), \\ S_{i,j-1} + s(-, b_j) \end{cases}$$

Proof:

$$\begin{array}{ccc} \cdots a_i & \cdots a_i & \cdots - \\ \cdots b_j & \cdots - & \cdots b_j \end{array}$$

Global Alignment Needlman-Wunsch

Given:

Scoring matrix

Sequences: $\mathbf{a} = a_1 a_2 \cdots a_n$; $\mathbf{b} = b_1 b_2 \cdots b_m$.

Matrix: $s(x, y)$ and gap penalty $s(x, -) = s(-, x) = g(x)$

Find:

$$S(\mathbf{a}, \mathbf{b}) = max \sum_{i=1}^{L} s(a_i^*, b_i^*)$$

Solution:

1. Define $S_{ij} = S(a_1 a_2 \cdots a_n, b_1 b_2 \cdots b_m)$ and set $S_{00} = 0$, $S_{0j} = \sum_{k=1}^{j} s(-, b_k)$, and $S_{i0} = \sum_{k=1}^{i} s(a_k, -)$

2.

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j), \\ S_{i-1,j} + s(a_i, -), \\ S_{i,j-1} + s(-, b_j) \end{cases}$$

Proof:

$$\begin{array}{ccc} \cdots a_i & \cdots a_i & \cdots - \\ \cdots b_j & \cdots - & \cdots b_j \end{array}$$

Global Alignment Needlman-Wunsch

Given:

Sequences: $\mathbf{a} = a_1 a_2 \cdots a_n$; $\mathbf{b} = b_1 b_2 \cdots b_m$.

Scoring matrix

$\boxed{\text{Matrix: } s(x,y)}$ and gap penalty $\boxed{s(x,-) = s(-,x) =}$ $g(x)$

Gap penalty

Find:

$$S(\mathbf{a},\mathbf{b}) = max \sum_{i=1}^{L} s(a_i^*, b_i^*)$$

Solution:

1. Define $S_{ij} = S(a_1 a_2 \cdots a_n, b_1 b_2 \cdots b_m)$
and set $S_{00} = 0$,
$S_{0j} = \sum_{k=1}^{j} s(-, b_k)$, and $S_{i0} = \sum_{k=1}^{i} s(a_k, -)$
2.

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j), \\ S_{i-1,j} + s(a_i, -), \\ S_{i,j-1} + s(-, b_j) \end{cases}$$

Proof:

$$\begin{array}{ccc} \cdots a_i & \cdots a_i & \cdots - \\ \cdots b_j & \cdots - & \cdots b_j \end{array}$$

Global Alignment Needlman-Wunsch

Given:

Sequences: $\mathbf{a} = a_1 a_2 \cdots a_n$; $\mathbf{b} = b_1 b_2 \cdots b_m$.

Scoring matrix

Matrix: $s(x, y)$ and gap penalty $s(x, -) = s(-, x) =$ $g(x)$

Gap penalty

Find:

$$S(\mathbf{a}, \mathbf{b}) = max \sum_{i=1}^{L} s(a_i^*, b_i^*)$$

Solution:

1. Define $S_{ij} = S(a_1 a_2 \cdots a_n, b_1 b_2 \cdots b_m)$

and set $S_{00} = 0$,
$S_{0j} = \sum_{k=1}^{j} s(-, b_k)$, and $S_{i0} = \sum_{k=1}^{i} s(a_k, -)$

Boundary conditions

2.

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j), \\ S_{i-1,j} + s(a_i, -), \\ S_{i,j-1} + s(-, b_j) \end{cases}$$

Proof:

$$\begin{array}{ccc} \cdots a_i & \cdots a_i & \cdots - \\ \cdots b_j & \cdots - & \cdots b_j \end{array}$$

Global Alignment Needlman-Wunsch

Given:

Sequences: $\mathbf{a} = a_1 a_2 \cdots a_n$; $\mathbf{b} = b_1 b_2 \cdots b_m$.

Scoring matrix

Matrix: $s(x, y)$ and gap penalty $s(x, -) = s(-, x) = g(x)$

Gap penalty

Find:

$$S(\mathbf{a}, \mathbf{b}) = max \sum_{i=1}^{L} s(a_i^*, b_i^*)$$

Solution:

1. Define $S_{ij} = S(a_1 a_2 \cdots a_n, b_1 b_2 \cdots b_m)$

and set $S_{00} = 0$,
$S_{0j} = \sum_{k=1}^{j} s(-, b_k)$, and $S_{i0} = \sum_{k=1}^{i} s(a_k, -)$

Boundary conditions

2.

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j), \\ S_{i-1,j} + s(a_i, -), \\ S_{i,j-1} + s(-, b_j) \end{cases}$$

Forward propagation

Proof:

$$\begin{array}{ccc} \cdots a_i & \cdots a_i & \cdots - \\ \cdots b_j & \cdots - & \cdots b_j \end{array}$$

$s(x,x)=2$	MATCH
$s(x,y)=-1 \text{ for } x \neq y$	MISMATCH
$s(x,-)=s(-,x)=-2$	GAP PENALTY

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j), \\ S_{i-1,j} + s(a_i, -), \\ S_{i,j-1} + s(-, b_j) \end{cases}$$

Score of the optimal alignment that ends at (i,j)

$s(x,x) = 2$ MATCH

$s(x,y) = -1$ for $x \neq y$ MISMATCH

$s(x,-) = s(-,x) = -2$ GAP PENALTY

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j), \\ S_{i-1,j} + s(a_i, -), \\ S_{i,j-1} + s(-, b_j) \end{cases}$$

			j-1	*j*	
		G	A	A	T
	G	▨	▨	▨	
i-1	G	▨	1	-1	
i	C	▨	4	?	
	T				

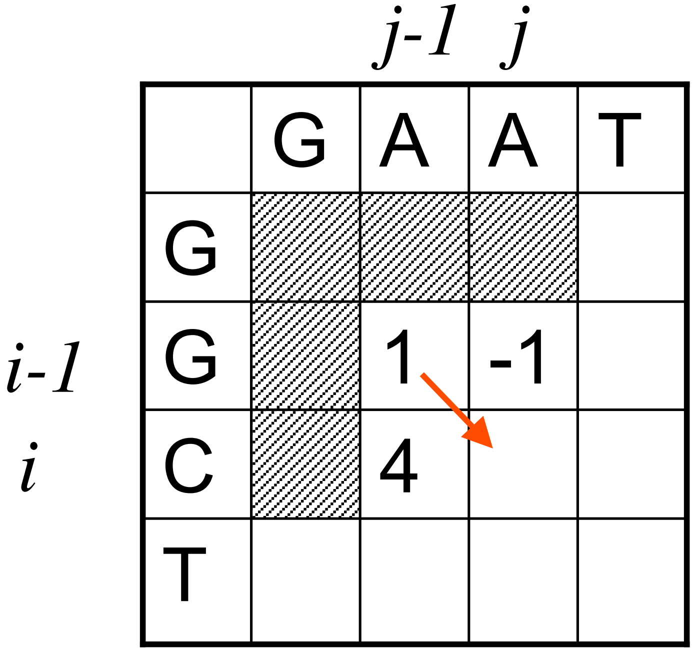
$s(x,x) = 2$ MATCH

$s(x,y) = -1$ for $x \neq y$ MISMATCH

$s(x,-) = s(-,x) = -2$ GAP PENALTY

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j), \\ S_{i-1,j} + s(a_i, -), \\ S_{i,j-1} + s(-, b_j) \end{cases}$$

Match/mismatch



			j-1	j	
		G	A	A	T
	G	▨	▨	▨	
i-1	G	▨	1	-1	
i	C	▨	4		
	T				

$\ldots.a^*_{i-1}$ C

$\ldots.b^*_{i-1}$ A

$s(x,x) = 2$ MATCH

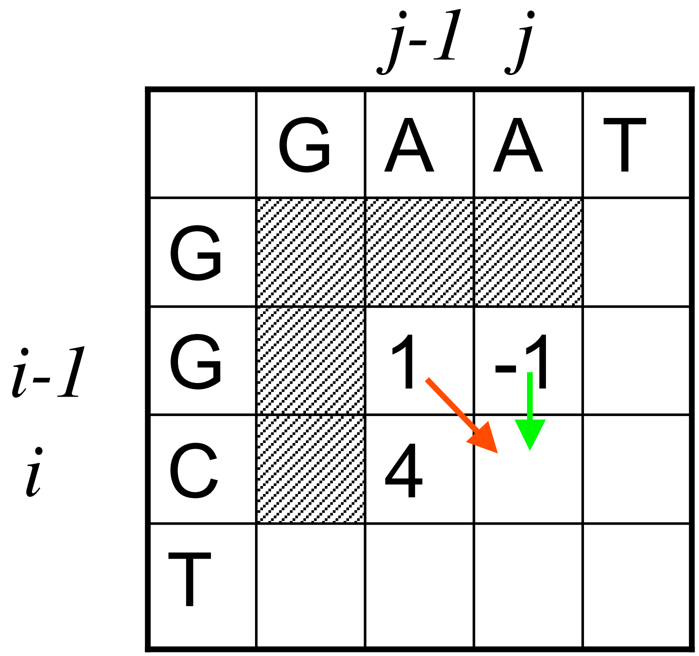
$s(x,y) = -1$ for $x \neq y$ MISMATCH

$s(x,-) = s(-,x) = -2$ GAP PENALTY

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j), \\ S_{i-1,j} + s(a_i, -), \\ S_{i,j-1} + s(-, b_j) \end{cases}$$

Match/mismatch

Gap in sequence #2



			j-1	j	
		G	A	A	T
	G	▨	▨	▨	
i-1	G	▨	1	-1	
i	C	▨	4		
	T				

....a^*_{i-1} C
....b^*_{i-1} A

....a^*_{i-1} C
....b^*_{i-1} --

$s(x,x) = 2$ MATCH

$s(x,y) = -1$ for $x \neq y$ MISMATCH

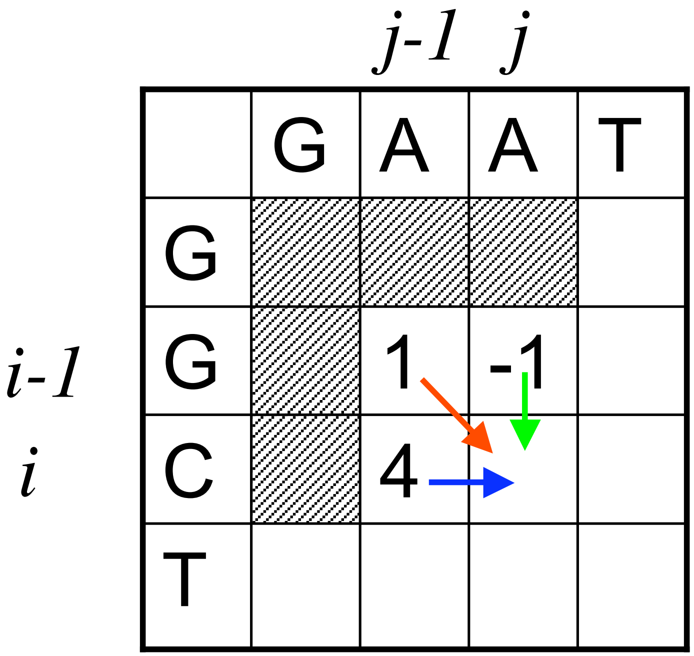
$s(x,-) = s(-,x) = -2$ GAP PENALTY

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j), \\ S_{i-1,j} + s(a_i, -), \\ S_{i,j-1} + s(-, b_j) \end{cases}$$

Match/mismatch

Gap in sequence #2

Gap in sequence #1



	G	A	A	T
G	▨	▨	▨	
G	▨	1	-1	
C	▨	4		
T				

….a^{*}_{i-1} C
….b^{*}_{i-1} A

….a^{*}_{i-1} C
….b^{*}_{i-1} --

….a^{*}_{i-1} --
….b^{*}_{i-1} A

$s(x,x) = 2$ MATCH

$s(x,y) = -1$ for $x \neq y$ MISMATCH

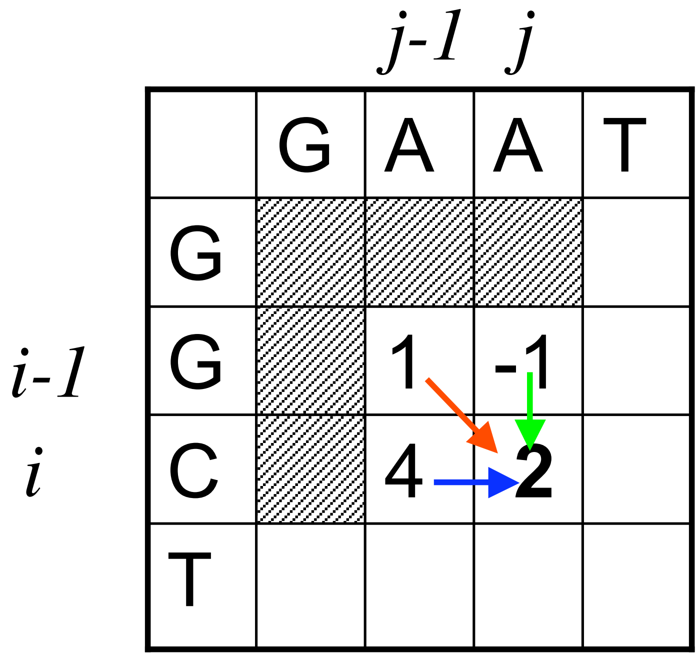
$s(x,-) = s(-,x) = -2$ GAP PENALTY

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j), \\ S_{i-1,j} + s(a_i, -), \\ S_{i,j-1} + s(-, b_j) \end{cases}$$

Match/mismatch

Gap in sequence #2

Gap in sequence #1



	G	A	A	T
G	▨	▨	▨	
G	▨	1	-1	
C	▨	4	2	
T				

....a^*_{i-1} C
....b^*_{i-1} A

1-1=0

....a^*_{i-1} C
....b^*_{i-1} --

-1-2=-3

2

....a^*_{i-1} --
....b^*_{i-1} A

4-2=2

Example

Algorithm

1. Build Sij matrix
2. Trace it back

Memory: O(NM)

Time: O(NM)

- There are EXACT algorithms that take less memory O(N).
- There are APPROXIMATE algorithms that take less time O(kN) or aN+O(pN).

Global Alignment (cont)

Arbitrary form of gap penalty

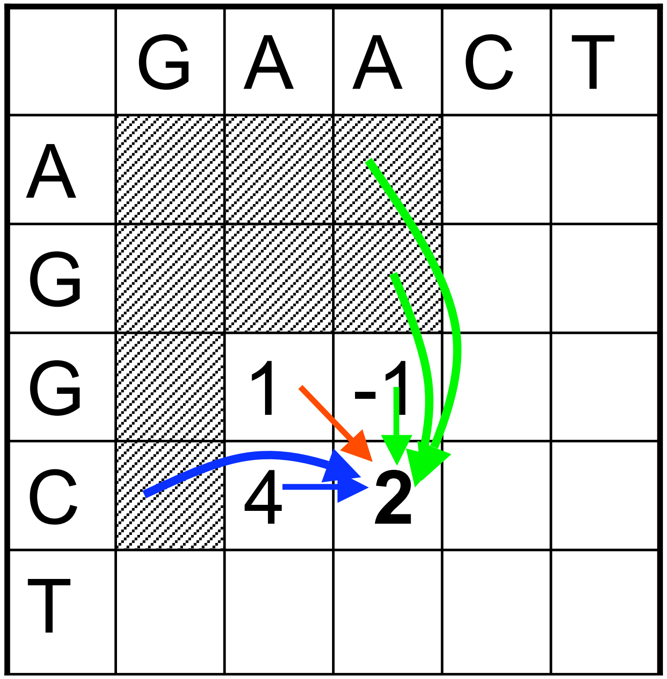
$g(k)$, where k is the length of the gap

Solution:

$S_{00} = 0$, $S_{0j} = g(j)$, and $S_{i0} = g(i)$

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j), \\ \max_{1 \le k \le j} \{S_{i-k,j} + g(k)\}, \\ \max_{1 \le l \le i} \ \{S_{i,j-l} + g(l)\} \end{cases}$$

Computation time $O(n^3)$

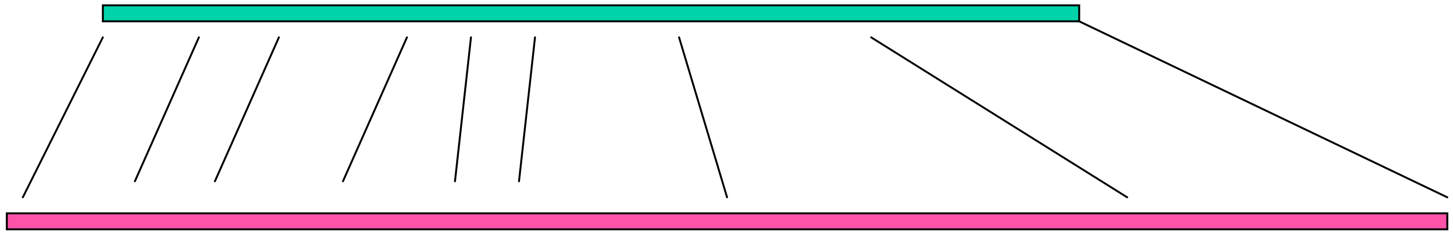


	G	A	A	C	T
A	▨	▨	▨		
G	▨	▨	▨		
G	▨	1	-1		
C	▨	4	**2**		
T					

Boundary conditions

- Global
- Local
- Global-local

Global alignment



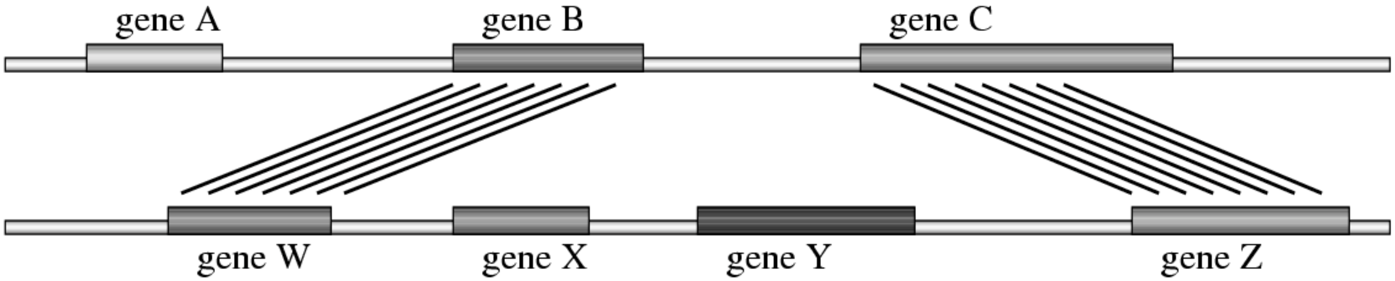
and set $S_{00} = 0$,

$S_{0j} = \sum_{k=1}^{j} s(-, b_k)$, and $S_{i0} = \sum_{k=1}^{i} s(a_k, -)$

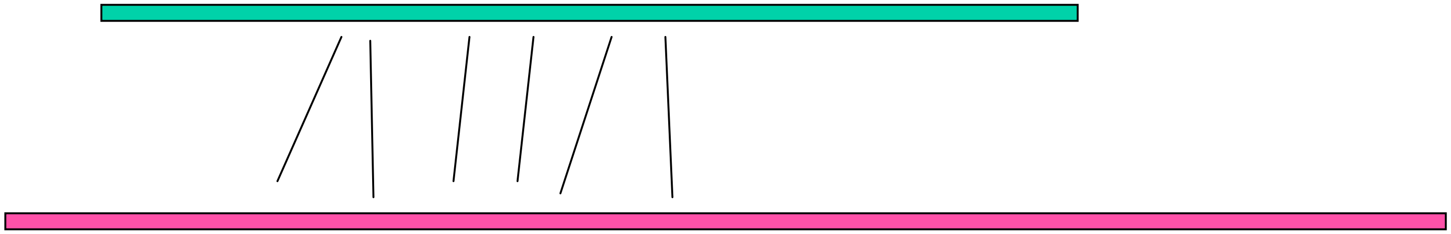
```
GGATCC..
---AAT..
```

Gap penalty at the head

Local Alignment



IDEA: No penalty for gaps at the ends



Local Alignment

Local Alignment Smith-Waterman

No penalties for head/tail gaps!

Solution:

$S_{00} = S_{0j} = S_{i0} = 0$

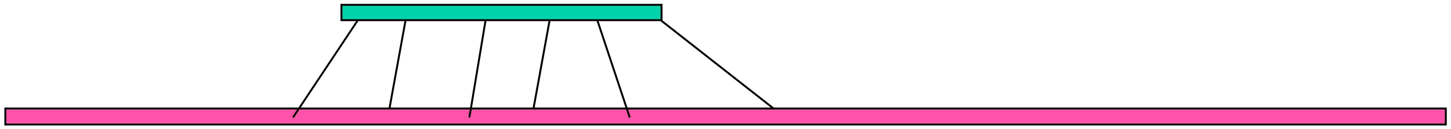
$$S_{ij} = \max \begin{cases} 0 \\ S_{i-1,j-1} + s(a_i, b_j), \\ \max_{1 \leq k \leq j} \{S_{i-k,j} + g(k)\}, \\ \max_{1 \leq l \leq i} \{S_{i,j-l} + g(l)\} \end{cases}$$

Start trace-back from $\max\{S_{ij}\}$

```
(GGA)---TCCAGT-----(ATTC)
        ---TCC-GT-----(GGCC)
```

Global-local Alignment

IDEA: No penalty for gaps at the ends of ONE SEQUENCE



Fitting one sequence into another (Global-Local Alignment)

No penalties for head/tail gaps for one sequence

Solution:

$S_{00} = S_{0j} = 0$, but $S_{i0} = g(i)$

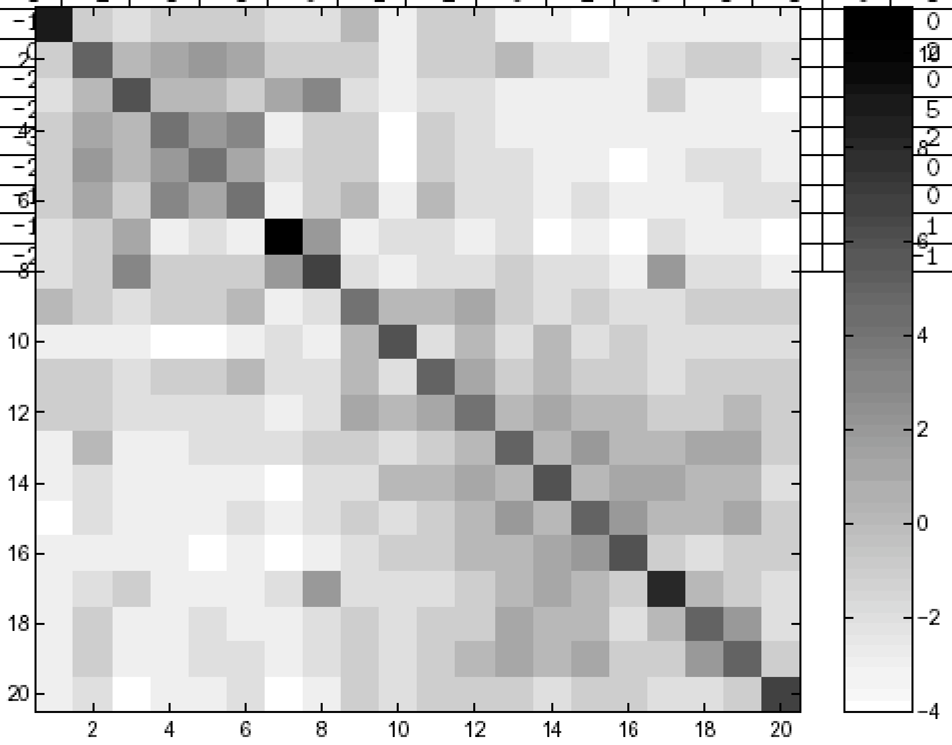
$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j), \\ \max_{1 \le k \le j} \{S_{i-k,j} + g(k)\}, \\ \max_{1 \le l \le i} \ \{S_{i,j-l} + g(l)\} \end{cases}$$

Start trace-back from $\max\{S_{nj}\}$

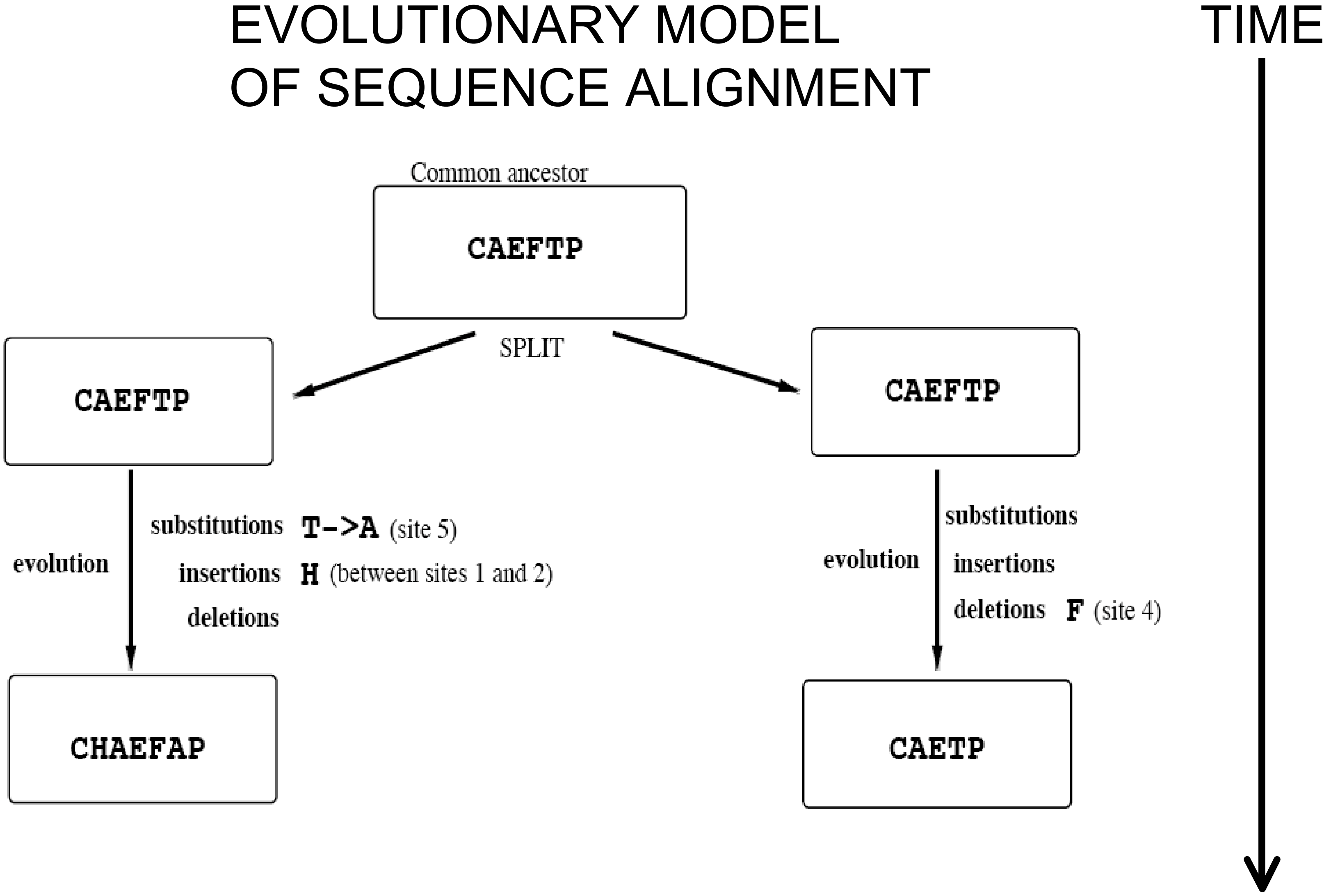
Scoring/substitution matrices

Realistic Scoring

	C	M	F	I	L	V	W	Y	A	G	T	S	Q	N	E	D	H
C	9	-1	-2	-1	-1	-1	-2	-2	0	-3	-1	-1	-3	-3	-4	-3	-3
M	-1	5	0	1	2	1	-1	-1	-1	-3	-1	-1	0	-2	-2	-3	-2
F	-2	0	6	0	0	-1	1	3	-2	-3	-2	-2	-3	-3	-3	-3	-1
I	-1	1	0	4	2	3	-3	-1	-1	-4	-1	-2	-3	-3	-3	-3	-3
L	-1	2	0	2	4	1	-2	-1	-1	-4	-1	-2	-2	-3	-3	-4	-3
V	-1	1	-1	3	1	4	-3	-1	0	-3	0	-2	-2	-3	-2	-3	-3
W	-2	-1	1	-3	-2	-3	11	2	-3	-2	-2	-3	-2	-4	-3	-4	-2
Y	-2	-1	3	-1	-1	-1	2	7	-2	-3	-2	-2	-1	-2	-2	-3	2
A	0	-1	-2	-1	-1	0	-3	-2	4	0	0	1	-1	-2	-1	-2	-2
G	-3	-3	-3	-4	-4	-3	-2	-3	0	6	-2	0	-2	0	-2	-1	-2
T	-1	-1	-2	-1	-1	0	-2	-2	0	-2	5	1	-1	0	-1	-1	-2
S	-1	-1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	-1
Q	-3	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2	0	0
N	-3	-2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	1	1
E	-4	-2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	5	2	0
D	-3	-3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2	6	-1
H	-3	-2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	-1	8
R	-3	-1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	-2	0
K	-3	-1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1	-1	-1
P	-3	-2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-1	-1	-2



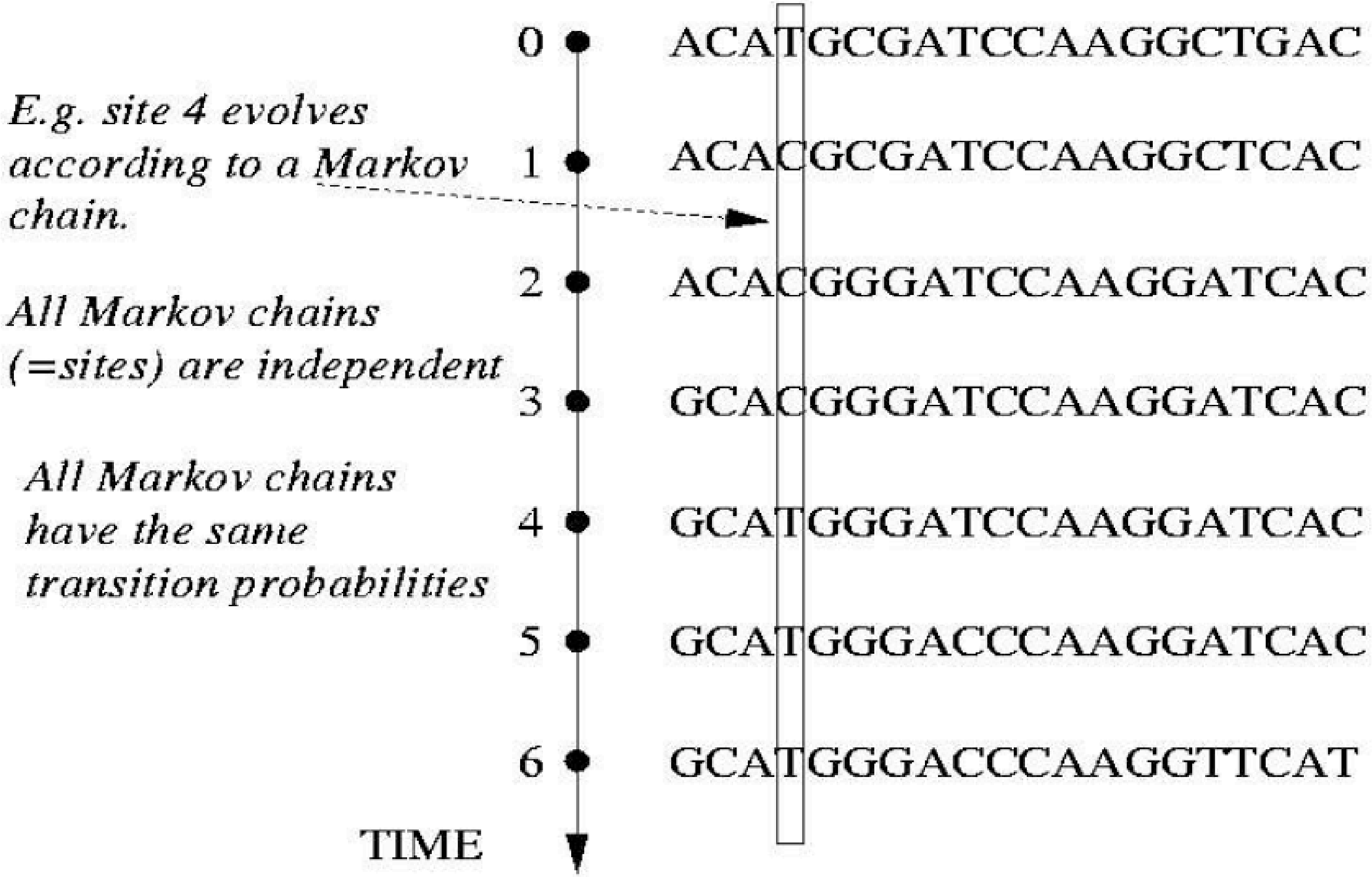
EVOLUTIONARY MODEL OF SEQUENCE ALIGNMENT



Evolutionarily correct alignment:

```
CHAEFAP
C-AE-TP
```

Models for sequence evolution (DNA): Each site of the DNA sequence evolves according to a Markov Chain with state space $\{\mathbf{A},\mathbf{C},\mathbf{G},\mathbf{T}\}$.



E.g. site 4 evolves according to a Markov chain.

All Markov chains (=sites) are independent

All Markov chains have the same transition probabilities

0 ACATGCGATCCAAGGCTGAC

1 ACACGCGATCCAAGGCTCAC

2 ACACGGGATCCAAGGATCAC

3 GCACGGGATCCAAGGATCAC

4 GCATGGGATCCAAGGATCAC

5 GCATGGGACCCAAGGATCAC

6 GCATGGGACCCAAGGTTCAT

MARKOV CHAIN

Let $X_0, X_1, X_2, X_3, \ldots$ be a Markov chain with **state space** S, for example $S = \{a, c, g, t\}$.

TRANSITION MATRIX

$$P = \begin{pmatrix} p_{a,a} & p_{a,c} & p_{a,g} & p_{a,t} \\ p_{c,a} & p_{c,c} & p_{c,g} & p_{c,t} \\ p_{g,a} & p_{g,c} & p_{g,g} & p_{g,t} \\ p_{t,a} & p_{t,c} & p_{t,g} & p_{t,t} \end{pmatrix}.$$

Here

$$p_{i,j} = \mathbf{P}(X_{n+1} = j | X_n = i)$$

for $n \geq 0$, where $i, j \in \{a, c, g, t\}$.

Simplest model for sequence evolution: Jukes-Cantor

$$\begin{pmatrix} p_{a,a} & p_{a,c} & p_{a,g} & p_{a,t} \\ p_{c,a} & p_{c,c} & p_{c,g} & p_{c,t} \\ p_{g,a} & p_{g,c} & p_{g,g} & p_{g,t} \\ p_{t,a} & p_{t,c} & p_{t,g} & p_{t,t} \end{pmatrix} = \begin{pmatrix} 1-3\alpha & \alpha & \alpha & \alpha \\ \alpha & 1-3\alpha & \alpha & \alpha \\ \alpha & \alpha & 1-3\alpha & \alpha \\ \alpha & \alpha & \alpha & 1-3\alpha \end{pmatrix}$$

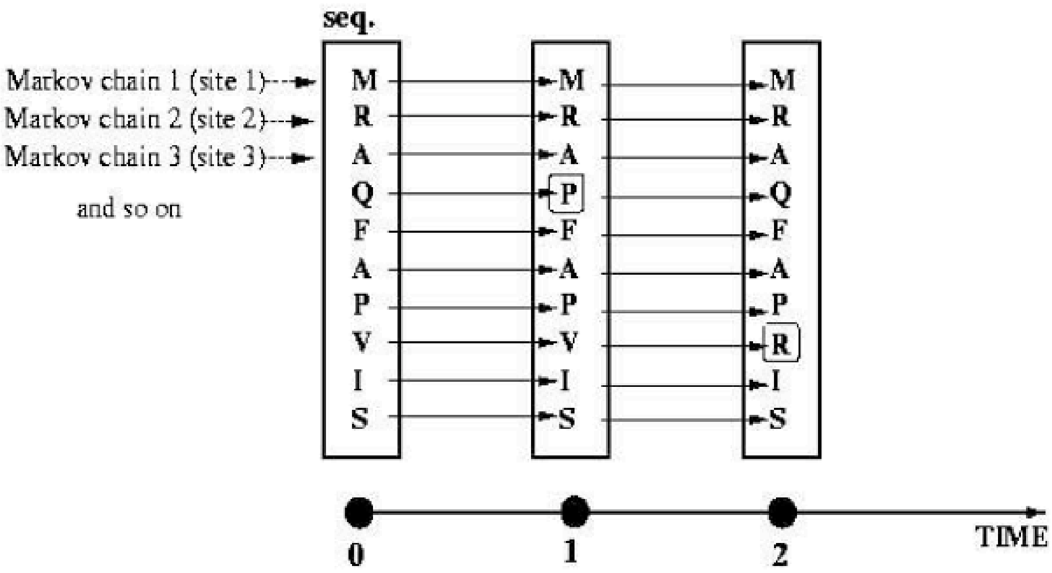
The stationary distribution is $\vec{\pi} = (0.25, 0.25, 0.25, 0.25)$.

The parameter α depends on the time scale
(if the unit time is 100.000 *generations,* α *would take a smaller value than if the unit time were chosen as* 200.000 *generations).*

Necessary: $\alpha < 1/3$.

The $n-$step transition probabilities can be computed:
$\mathbf{P}(X_n = i|X_0 = i) = 0.25 + 0.75 \cdot (1-4\alpha)^n$, for $i \in \{a, c, g, t\}$.
$\mathbf{P}(X_n = j|X_0 = i) = 0.25 - 0.25 \cdot (1-4\alpha)^n$, for $i, j \in \{a, c, g, t\}$, $i \neq j$.

Underlying model: Each site in the sequence evolves *according to a Markov chain*, and *independently* of the other sites.



All the Markov chains have the *same* transition matrix P (matrix with dimension 20×20).

FROM TRANSITION MATRIX TO ALIGNMENT SCORES

Two hypothesis:

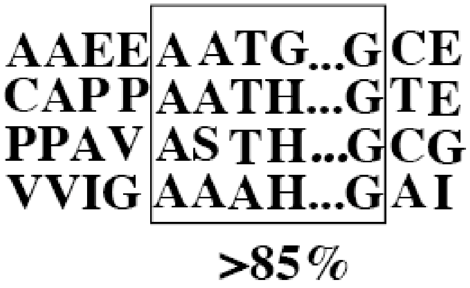
1. Sequences S1 and S2 are unrelated (=random matching)
2. Sequences S1 and S2 have a common ancestor.

Score = Log (P1/P2)

P1 - probability of observed alignment given model 1
P2 - probability of observed alignment given model 2

Dayhoff et al. (1978) used ungapped multiple alignments of certain well-conserved regions from closely related proteins.

(71 *groups of proteins, all in all* 1572 *changes.*)



```
AAEE AATG...G CE
CAPP AATH...G TE
PPAV ASTH...G CG
VVIG AAAH...G AI
     >85%
```

Dr. Margaret Oakley Dayhoff (1925-1983)

The most parsimonious tree

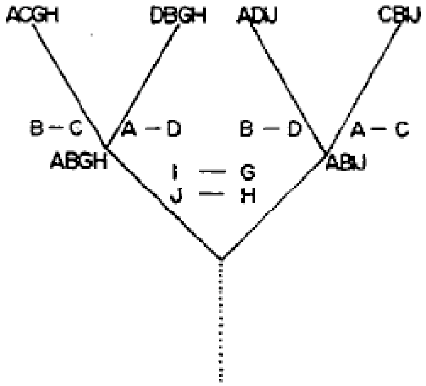
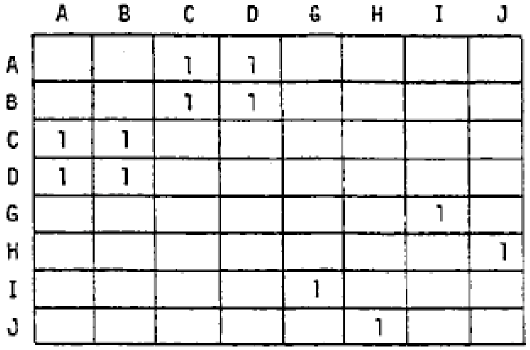


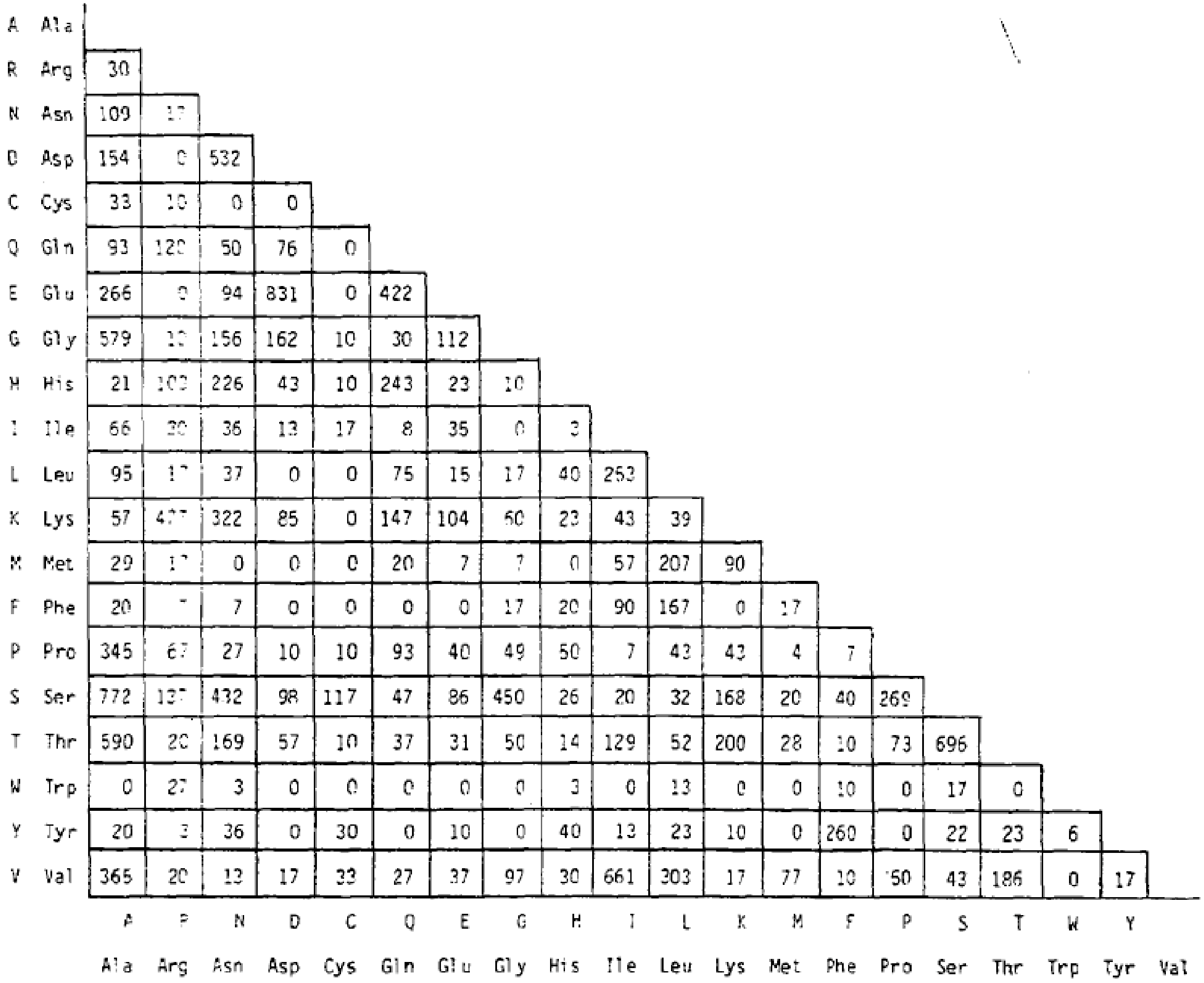
Figure 78. Simplified phylogenetic tree. Four "observed" proteins are shown at the top. Inferred ancestors are shown at the nodes. Amino acid exchanges are indicated along the branches.



	A	B	C	D	G	H	I	J
A			1	1				
B			1	1				
C	1	1						
D	1	1						
G							1	
H								1
I					1			
J						1		

Figure 79. Matrix of accepted point mutations derived from the tree of Figure 78.

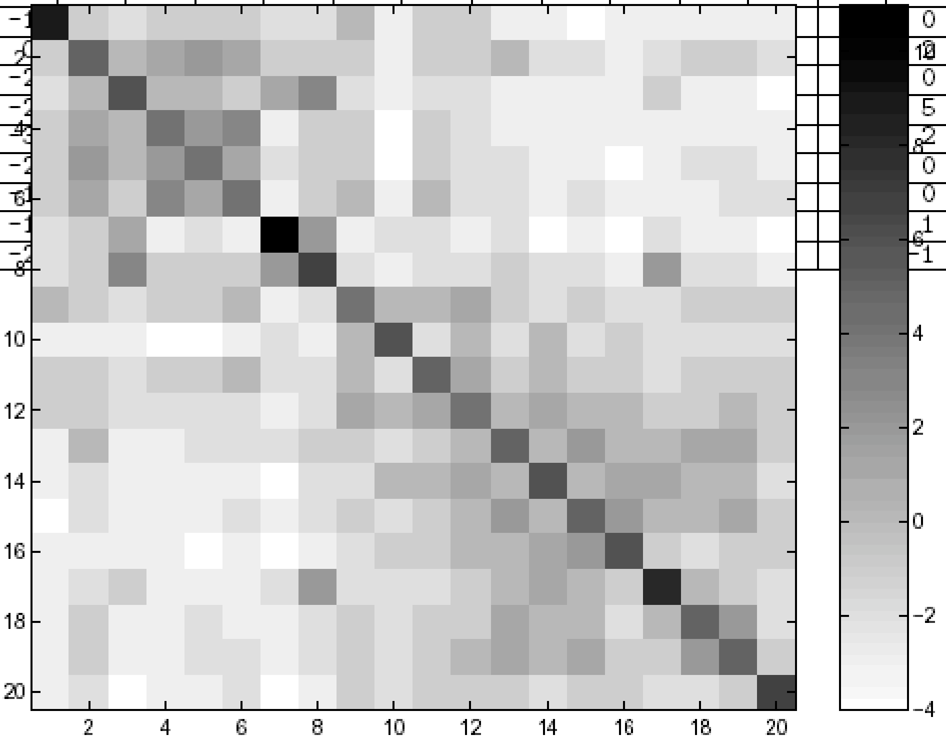
The number of accepted mutations (A_{ij})



		A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His	I Ile	L Leu	K Lys	M Met	F Phe	P Pro	S Ser	T Thr	W Trp	Y Tyr	Val
A	Ala																				
R	Arg	30																			
N	Asn	109	17																		
D	Asp	154	0	532																	
C	Cys	33	10	0	0																
Q	Gln	93	120	50	76	0															
E	Glu	266	0	94	831	0	422														
G	Gly	579	10	156	162	10	30	112													
H	His	21	103	226	43	10	243	23	10												
I	Ile	66	30	36	13	17	8	35	0	3											
L	Leu	95	17	37	0	0	75	15	17	40	253										
K	Lys	57	477	322	85	0	147	104	60	23	43	39									
M	Met	29	17	0	0	0	20	7	7	0	57	207	90								
F	Phe	20	4	7	0	0	0	0	17	20	90	167	0	17							
P	Pro	345	67	27	10	10	93	40	49	50	7	43	43	4	7						
S	Ser	772	137	432	98	117	47	86	450	26	20	32	168	20	40	269					
T	Thr	590	20	169	57	10	37	31	50	14	129	52	200	28	10	73	696				
W	Trp	0	27	3	0	0	0	0	0	3	0	13	0	0	10	0	17	0			
Y	Tyr	20	3	36	0	30	0	10	0	40	13	23	10	0	260	0	22	23	6		
V	Val	365	20	13	17	33	27	37	97	30	661	303	17	77	10	50	43	186	0	17	

Realistic Scoring

	C	M	F	I	L	V	W	Y	A	G	T	S	Q	N	E	D	H
C	9	-1	-2	-1	-1	-1	-2	-2	0	-3	-1	-1	-3	-3	-4	-3	-3
M	-1	5	0	1	2	1	-1	-1	-1	-3	-1	-1	0	-2	-2	-3	-2
F	-2	0	6	0	0	-1	1	3	-2	-3	-2	-2	-3	-3	-3	-3	-1
I	-1	1	0	4	2	3	-3	-1	-1	-4	-1	-2	-3	-3	-3	-3	-3
L	-1	2	0	2	4	1	-2	-1	-1	-4	-1	-2	-2	-3	-3	-4	-3
V	-1	1	-1	3	1	4	-3	-1	0	-3	0	-2	-2	-3	-2	-3	-3
W	-2	-1	1	-3	-2	-3	11	2	-3	-2	-2	-3	-2	-4	-3	-4	-2
Y	-2	-1	3	-1	-1	-1	2	7	-2	-3	-2	-2	-1	-2	-2	-3	2
A	0	-1	-2	-1	-1	0	-3	-2	4	0	0	1	-1	-2	-1	-2	-2
G	-3	-3	-3	-4	-4	-3	-2	-3	0	6	-2	0	-2	0	-2	-1	-2
T	-1	-1	-2	-1	-1	0	-2	-2	0	-2	5	1	-1	0	-1	-1	-2
S	-1	-1													0	0	-1
Q	-3	0													2	0	0
N	-3	-2													0	1	1
E	-4	-2													5	2	0
D	-3	-3													2	6	-1
H	-3	-2													0	-1	8
R	-3	-1													0	-2	0
K	-3	-1													1	-1	-1
P	-3	-2													-1	-1	-2



Database searches

Problem:

Alignment of a gene 1000bp
against the Human genome $3 \cdot 10^9$bp
...10^{12} operations…

- protein against a database of 10^5 proteins
  -> 10^9 operations
- genome against genome
  -> 10^{14} operations

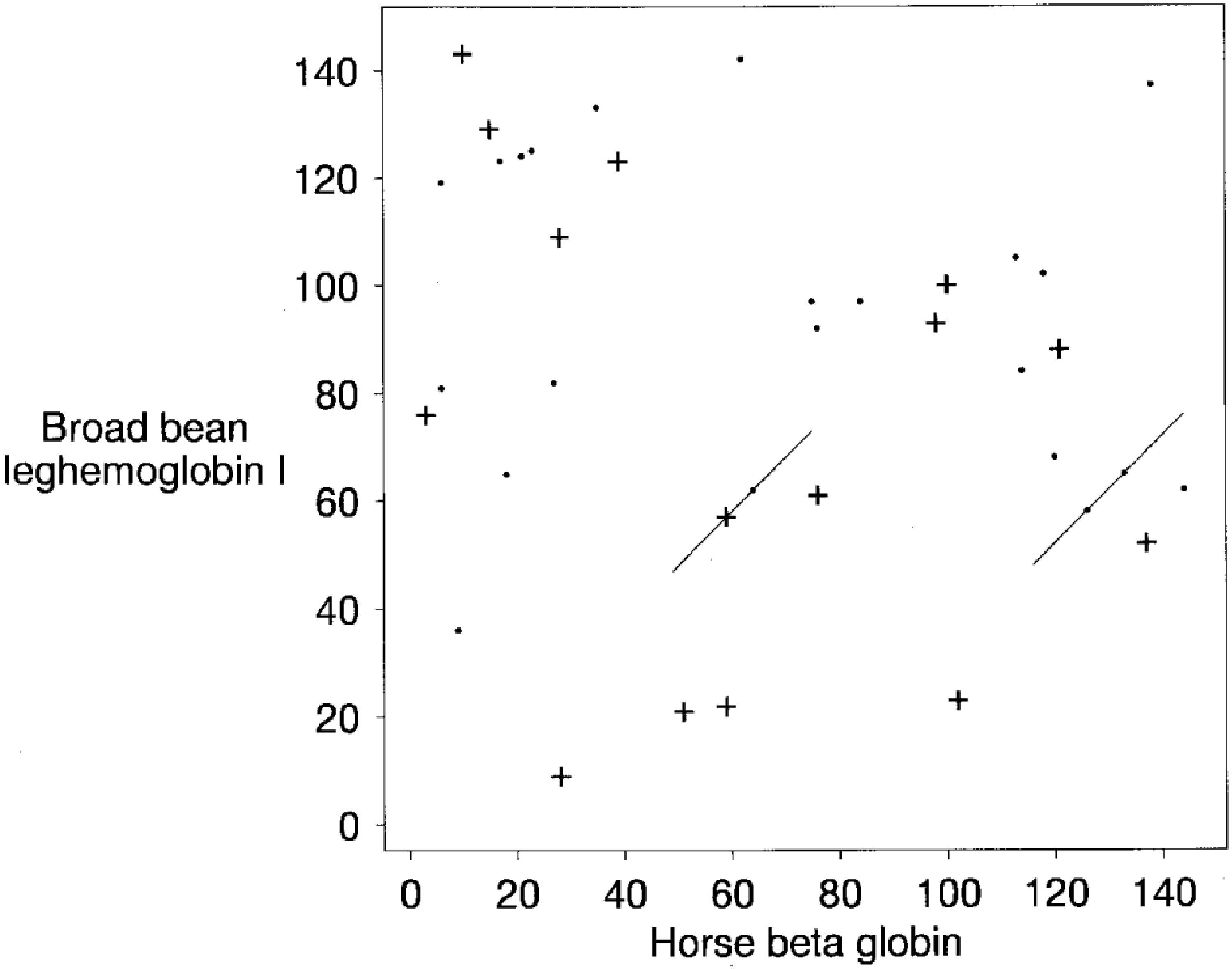
NEED FASTER ALGORITHMS

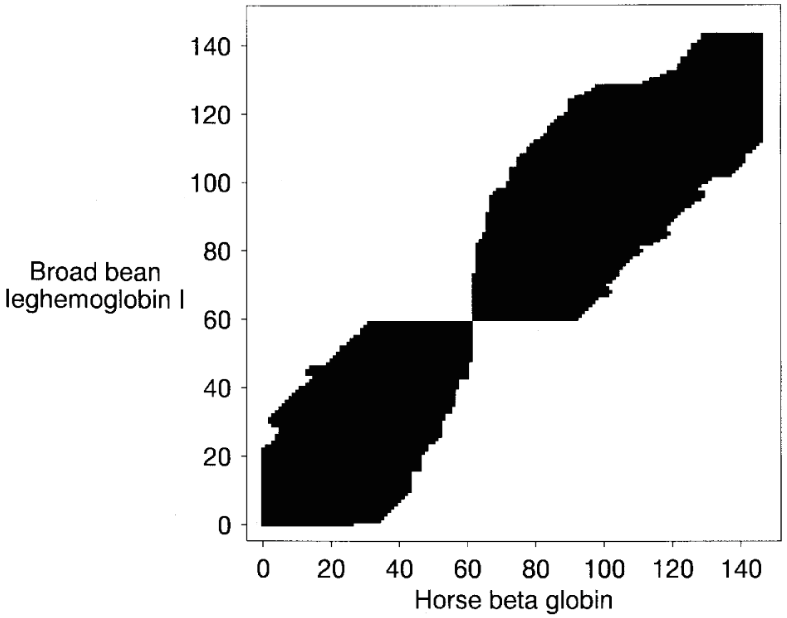
Database searches

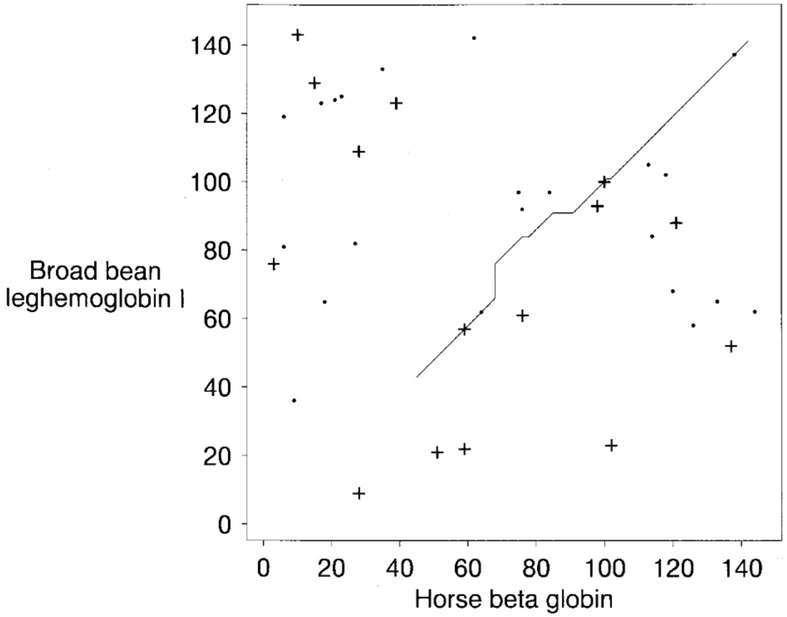
BLAST

- Pre-processing: Low Complexity Regions (LCRs)
- Scanning for common words (hits)
- Two-hit heuristic
- HSP (high-scoring segment pair) $> S$
- Constrained gaped extension
- E-value

WARNING: The final alignment is not very good!


140
120
100
80
60
40
20
0
Broad bean leghemoglobin I
0
20
40
60
80
100
120
140
Horse beta globin





```
Leghemoglobin  43 FSFLKDSAGVVDSPKLGAHAEKVFGMVRDSAVQLRATGEVV--LDGKDGS------  90
                  F  L +   V+ +PK+ AH +KV          L + GE V  LD   G+
Beta globin    45 FGDLSNPGAVMGNPKVKAHGKKV----------LHSFGEGVHHLDNLKGTFAALSE  90


Leghemoglobin  91 IHIQKGVLDP-HFVVVKEALLKTIKEASGDKWSEELSAAWEVAYDGLATAI 140
                  +H  K  +DP +F ++   L+  +    G  ++ EL A+++    G+A A+
Beta globin    91 LHCDKLHVDPENFRLLGNVLVVVLARHFGKDFTPELQASYQKVVAGVANAL 141
```