BIOINFORMATIK II
PROBABILITY & STATISTICS
Summer semester 2007

The University of Zürich and ETH Zürich

Lecture 4: Evolutionary models and substitution matrices (PAM and BLOSUM).

Prof. Andrew Barbour
Dr. Béatrice de Tilière

Adapted from a course by
Dr. D. Schuhmacher & Dr. D. Svensson.

Previous lecture:

Theory of Markov chains and applications in different models.

Usually, we modelled the development over different positions in one DNA sequence (states=nucleotides, index/"time"=position in the sequence).

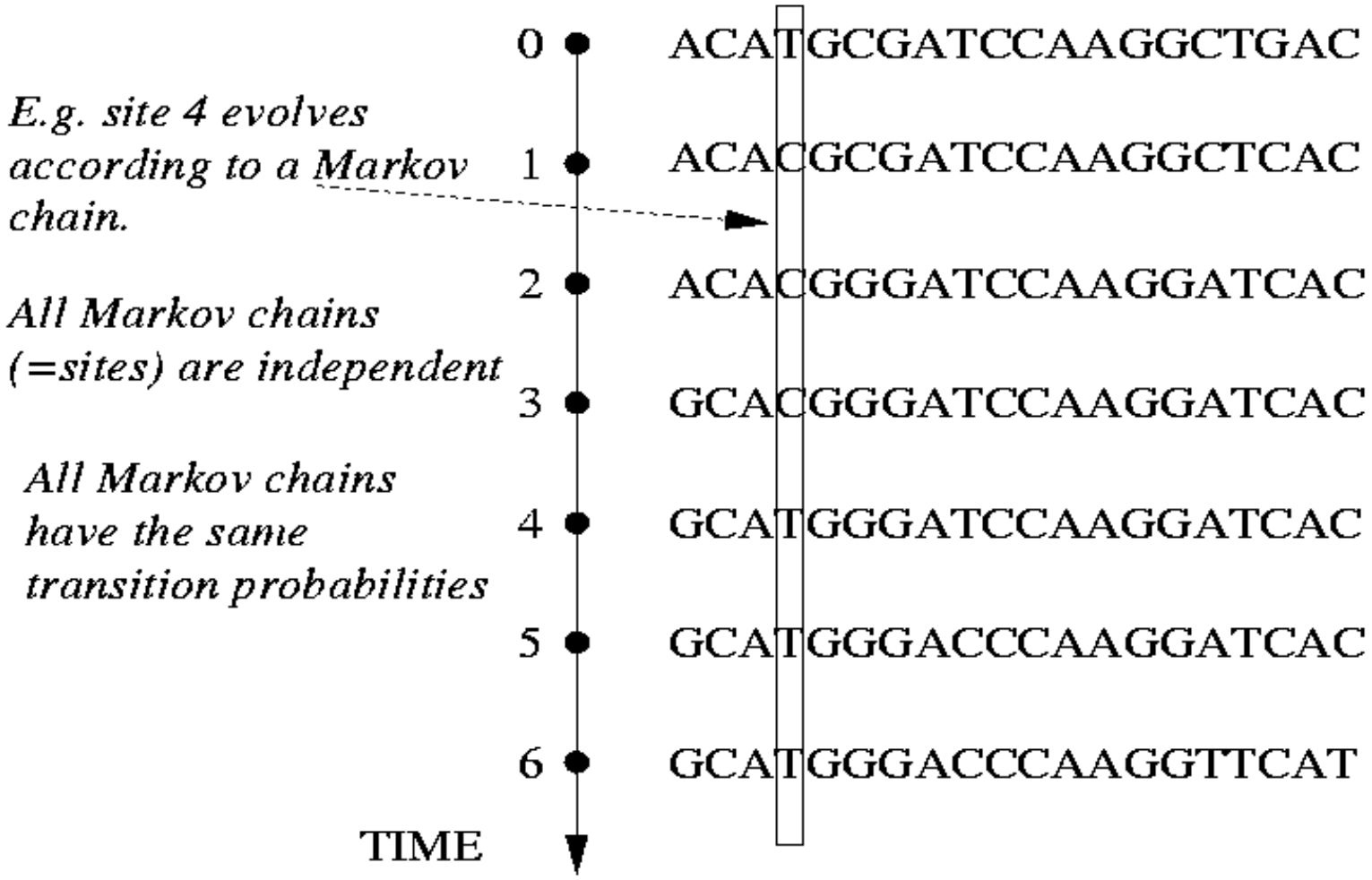
This lecture:

We use Markov chains to model the development of individual positions in a DNA (or Protein) sequence over time (states=nucleotides/amino acids, index/"time"=time in which the sequence evolves.

Objectives for today:

- studying models for sequence evolution
- deriving "good" substitution matrices, i.e. matrices that give a "realistic" score to each possible substitution in a sequence (DNA or protein).

Models for sequence evolution (DNA): Each site of the DNA sequence evolves according to a Markov Chain with state space {**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 $i, j \in \{a, c, g, t\}$, $i \neq j$.

The Jukes-Cantor model is not entirely realistic (*e.g. all kinds of substitutions are equally likely to occur*).

More complicated and more realistic is for example the Kimura model:

$$P = \begin{pmatrix} 1-\alpha-2\beta & \beta & \alpha & \beta \\ \beta & 1-\alpha-2\beta & \beta & \alpha \\ \alpha & \beta & 1-\alpha-2\beta & \beta \\ \beta & \alpha & \beta & 1-\alpha-2\beta \end{pmatrix}$$

Necessary: $\alpha + 2\beta < 1$. Stationary distr.: $\vec{\pi} = (0.25, 0.25, 0.25, 0.25)$.

$\alpha =$ prob. for *'transitions'* (purine to purine or pyrimidine to pyrimidine).

$\beta =$ prob. for *'transversions'* (purine to pyrimidine or vice versa)

(purines: `a,g` / pyrimidines: `c,t`)

Comments:

There are even more realistic Markov models for DNA substitution (e.g. *Hasegawa, Kishino, Yano*, and many others).

Most models assume that sites evolve independently (which is not entirely realistic). Some models allow different sites to evolve at different rates.

Why not use more realistic models ? Because they are very difficult to handle! The more complicated the model we use, the more complicated it is to compute the probabilities of interest (and, the more complicated the model we use, the more parameters we need to estimate).

The simpler ones seem to give reasonably sensible results.

Always required in mathematical modeling: a balance between *realism* and *mathematical tractability*.

Evolutionary models for proteins?

Similar, but here it is much more important to account for the wide range of different transition probabilities associated with amino acid substitutions.

$\longrightarrow$ See the construction of substitution matrices!

Substitution matrices

Scoring systems

Very important in bioinformatics: Comparisons of sequences, DNA *or* protein (e.g. for inferring molecular function by finding similarity to a sequence with known function).

For that purpose needed: "Good alignment" of sequences.

For that purpose needed: A measure for judging the quality of an alignment in relation to other possible alignments, a **scoring system**.

We use additive scoring systems:

Look at each position of a given alignment, and assign a score for the "quality of the match" at this position (forget about gaps for now). The total (or *cumulative*) score is obtained by adding the scores for the individual positions.

Simple example: Two DNA sequences; score for a match: $+1$, score for a mismatch -1.

E.g.: `a a g t t t c t t g`

`a a a c t c c c t g`

Individual scores: 1 1 -1 -1 1 -1 1 -1 1 1

$\Longrightarrow$ Cumulative score: $6 - 4 = 2$

Maybe more realistic: score for a match: $+1$, score for a transition: $-1/2$, score for a transversion: -1. Cumulative score in that case: $6 - 2 = 4$.

Scoring matrices

The scores for the individual positions can be displayed in a so-called **substitution matrix** (also called **scoring matrix**). This is a usually symmetrical 4×4 (DNA) resp. 20×20 (protein) matrix which has as entry (i, j) the score that we assign if at a position the nucleotides resp. the amino acids i and j are aligned.

E.g. for the second example from the last slide:

$$S = \begin{pmatrix} s_{a,a} & s_{a,c} & s_{a,g} & s_{a,t} \\ s_{c,a} & s_{c,c} & s_{c,g} & s_{c,t} \\ s_{g,a} & s_{g,c} & s_{g,g} & s_{g,t} \\ s_{t,a} & s_{t,c} & s_{t,g} & s_{t,t} \end{pmatrix} = \begin{pmatrix} 1 & -1 & -1/2 & -1 \\ -1 & 1 & -1 & -1/2 \\ -1/2 & -1 & 1 & -1 \\ -1 & -1/2 & -1 & 1 \end{pmatrix}$$

How can we find a biologically sensible scoring matrix?

For DNA sequences: simple scoring matrices (like the one presented) are often effective.

$\longrightarrow$ Usually, no need to worry.

For protein sequences: some substitutions are clearly more likely to occur than others (presumably due to similar chemical properties of the amino acids involved); e.g. isoleucine for valine, serine for threonine, so-called *conservative substitutions*.

We get considerably better alignments if we take this into account.

$\longrightarrow$ Use scoring matrices that are derived by statistical analysis of protein data.

Biologically sensible scoring matrices for proteins

Specifications:

- *identical amino acids should be given greater score than any substitution;*
- *conservative substitutions should be given greater score than non-conservative ones;*
- *different sets of values* may be desired for comparing very similar sequences (*e.g. homologies in mouse and rat*) as opposed to highly divergent sequences (*e.g. homologies in mouse and yeast*); i.e. *we usually want our scoring matrices to take into account the evolutionary distance between the sequences involved!*

Scoring matrices, overview

There are two frequently used approaches to finding substitution matrices. They lead to

1) **the PAM family of substitution matrices**

   Main concepts used:

   Markov chains and phylogenetic trees

   (for "fitting" an evolutionary model)

   log-likelihood ratios

   (for getting a scoring matrix from an estimated transition matrix)

2) **the BLOSUM family of substitution matrices**

   Main concept used:

   log-likelihood ratios

   (for getting a scoring matrix from a matrix of estimated substitution probabilities)

The PAM family of substitution matrices

(Dayhoff, Schwartz, and Orcutt, 1978)

It requires the use of

- Markov chains and phylogenetic trees
  (for "fitting" an evolutionary model)
- log-likelihood ratios
  (for getting a scoring matrix from an estimated transition matrix)

"PAM" = **P**oint (or **P**ercentage) **A**ccepted **M**utations.

***

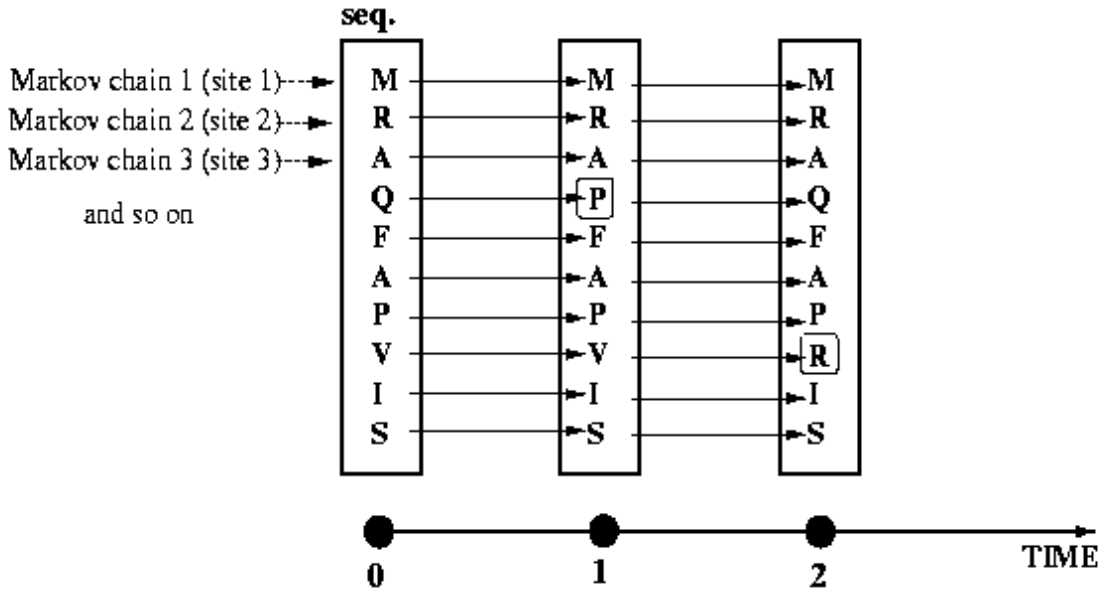
"accepted mutations" = those mutations able to spread in the population and become dominating (typically mutations that do not disrupt the protein function).

PAM matrices

In fact, there are two types of matrices involved here:

- a PAM **Markov transition** matrix
  (= *the table of estimated transition probabilities for the underlying evolutionary model*);
- a PAM **substitution** matrix
  (= *the table of scores for all possible pairs of amino acids*).

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).

Dayhoff et al. (1978) *estimated* the one-step transition matrix P from protein sequence data. How...? $\longrightarrow$

Construction of a PAM1 transition matrix

A **PAM1 transition matrix** is the Markov transition matrix applying for a time period over which we expect 1% *of the amino acids to undergo accepted point mutations*. The steps involved in the estimation:

- *Align protein sequences* that are at least 85 % identical.
- Reconstruct phylogenetic trees and *infer ancestral sequences*.
- *Count the amino acid replacements* that occurred along the trees (i.e. count mutations accepted by natural selection).
- Use these counts to *estimate probabilities* for the replacements.

The first step was to find reliable data.

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
CAP P AATH...G TE
PPAV AS TH...G CG
VVIG AAAH...G A I

>85%

In any block, any two sequences did not differ more than 15%.
(*The idea was to keep the number of sites that have encountered several changes low.*)

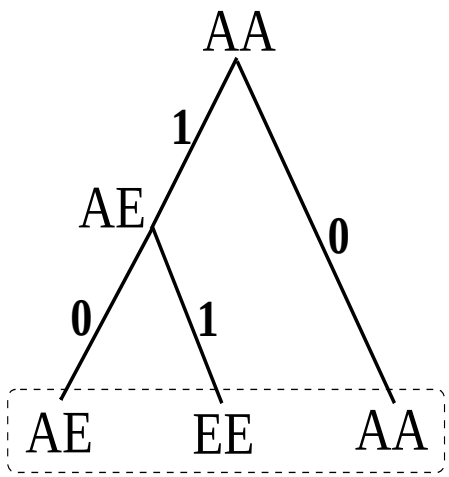
These aligned regions then were used to infer the underlying evolutionary tree[s] (there might be more than one).

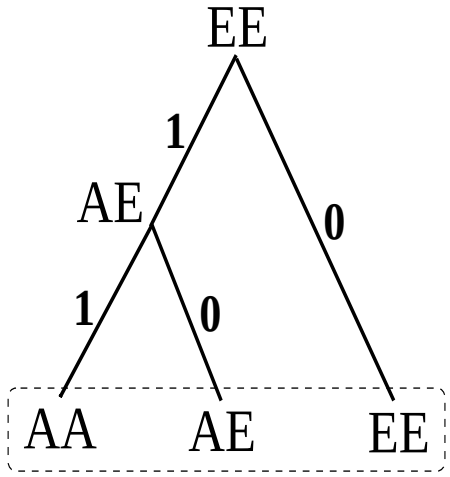
Maximum parsimony was used: ⟶

Maximum parsimony was used to infer the *underlying evolutionary tree[s]* and *ancestral sequences*.

A *most parsimonious* tree is a tree structure such that the total number of substitutions across the tree is minimal. Ex:

Data: seq1: **AA**
seq2: **AE**
seq3: **EE**





All kinds of amino acid substitutions that occurred along the tree[s] were then counted.

For example, in each tree above substitutions between A and E occurred 2 times.

Why do we use trees?

To avoid overcounting!

Our count might be biased by closely related sequences that are overrepresented in our database.

Trees $\Longrightarrow$ sequences are grouped in the "right" way (in general: very similar sequences succeed one another in the tree)

$\Longrightarrow$ we have mainly transitions between these sequences, and only a few transitions to other, more different sequences, so the corresponding substitutions do not get an unnatural importance.

Suppose that the amino acids are numbered from 1 to 20. (For simplicity, we assume that there was only one tree).

Let $A_{j,k}$ be the number of times substitutions from j to k were observed in the tree.

Result: a 'count' matrix.

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & ... & A_{1,19} & A_{1,20} \\ A_{2,1} & A_{2,2} & ... & A_{2,19} & A_{2,20} \\ ... & ... & ... & ... & ... \\ A_{19,1} & A_{19,2} & ... & A_{19,19} & A_{19,20} \\ A_{20,1} & A_{20,2} & ... & A_{20,19} & A_{20,20} \end{pmatrix}$$

This matrix was then used to *estimate a Markov transition matrix.*

How...? $\longrightarrow$

First, for any pair (j, k) define

$$a_{j,k} := \frac{A_{j,k}}{\sum_{m=1}^{20} A_{j,m}}.$$

This is the *observed relative frequency* for the substitution $j \to k$.

The $a_{j,k}$'s are *estimated probabilities.*

These probabilities were then *scaled* in a certain way:
For $j \neq k$,

$$p_{j,k} := c \cdot a_{j,k}$$

and

$$p_{j,j} := 1 - \sum_{k \neq j} c \cdot a_{j,k},$$

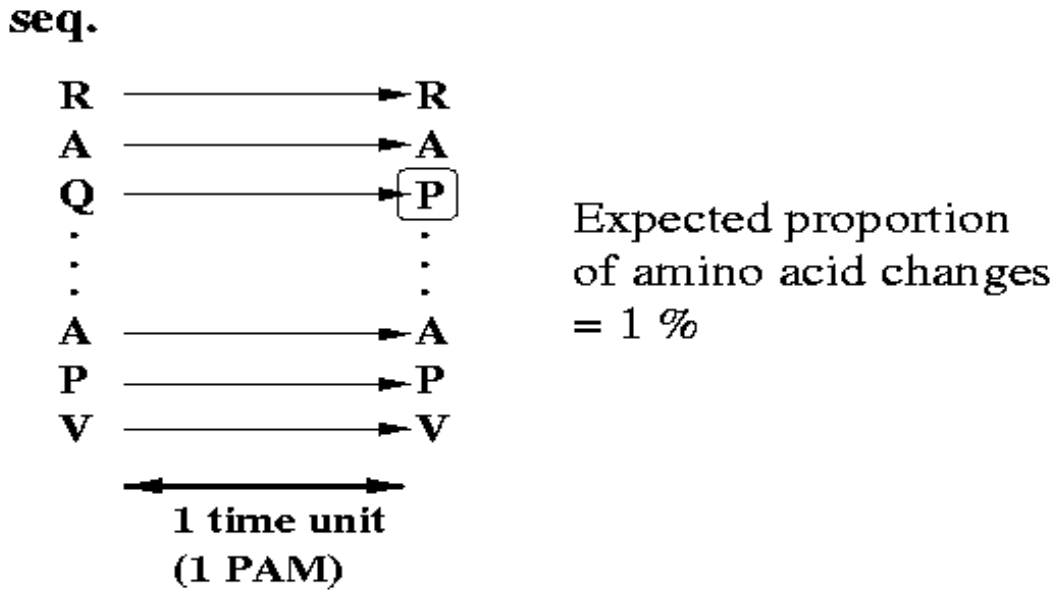
where the scaling constant c is sufficiently small so that
$p_{j,j} \geq 0$ for all j.

... but why the scaling factor c? $\longrightarrow$

Why the scaling factor c?

To account for the evolutionary distance!

Goal: to choose a value of c which renders a transition matrix that is *'useful for short evolutionary periods'*.
More exactly: choose a value of c, such that 1% *of the amino acids are expected to undergo accepted point mutations* during one time unit.



Such a time unit is called **an evolutionary distance of 1 PAM**.

How to choose the c?

To determine c, it *suffices to consider one of the sites* in the sequence, i.e. we consider only one of the parallel Markov chains.

Let $Z_n =$ *the amino acid present at the site considered at time* n, $n \geq 0$. (hence $1 \leq Z_n \leq 20$, since the AA's are coded as 1 to 20).

The probability that the site will change after 1 PAM time unit (i.e. after one step) is given by

$$\mathbf{P}(Z_1 \neq Z_0) = \sum_{j=1}^{20} \mathbf{P}(Z_0 = j, Z_1 \neq j)$$

$$= \sum_{j=1}^{20} \mathbf{P}(Z_1 \neq j | Z_0 = j) \cdot \mathbf{P}(Z_0 = j) \approx \sum_{j=1}^{20} \mathbf{P}(Z_1 \neq j | Z_0 = j) \cdot q_j,$$

where q_j is the *observed frequency of the amino acid no.* j in the original blocks of aligned proteins.

One wants the probability that the site will change after 1 PAM to be equal to 0.01. (*That implies an average change of* 1%.)

$$
\begin{aligned}
0.01 &= \sum_{j=1}^{20} \mathbf{P}(Z_1 \neq j | Z_0 = j) \cdot q_j \\
&= \sum_{j=1}^{20} \Big(\sum_{k \neq j} \mathbf{P}(Z_1 = k | Z_0 = j) \Big) \cdot q_j \\
&\approx \sum_{j=1}^{20} \Big(\sum_{k \neq j} p_{j,k} \Big) \cdot q_j \\
&= \sum_{j=1}^{20} \Big(\sum_{k \neq j} c \cdot a_{j,k} \Big) \cdot q_j \\
&= c \cdot \sum_{j=1}^{20} \sum_{k \neq j} q_j \cdot a_{j,k}.
\end{aligned}
$$

That is, we want

$$0.01 = c \cdot \sum_{j=1}^{20} \sum_{k \neq j} q_j \cdot a_{j,k}.$$

Therefore, using the estimated probabilities q_j and $a_{j,k}$, just put

$$c = \frac{0.01}{\sum_{j=1}^{20} \sum_{k \neq j} q_j \cdot a_{j,k}}.$$

Thus, with this choice for c, the *PAM transition matrix* is obtained ('one-step', i.e. for the evolutionary distance of 1 PAM) .

How can this transition matrix be turned into a scoring matrix?

How can the transition matrix be turned into a scoring matrix?

Consider two given protein sequences $\boldsymbol{s} = a_1 a_2 \cdots a_n$ and $\boldsymbol{s'} = b_1 b_2 \cdots b_n$ (at a evolutionary distance of 1 PAM, say).

The score for aligning $\boldsymbol{s}$ with $\boldsymbol{s'}$ is generated *by comparing two different hypothesis* H_0 and H_A:

- H_0: $\boldsymbol{s}$ *and* $\boldsymbol{s'}$ *are not evolutionarily related (i.e. a chance alignment).*
- H_A: $\boldsymbol{s}$ *and* $\boldsymbol{s'}$ *are evolutionarily related (i.e.* $\boldsymbol{s'}$ *depends on* $\boldsymbol{s}$ *via the Markov model).*

Under H_0, we have a chance alignment

$\boldsymbol{s}$: $a_1 a_2 \cdots a_n$

$\boldsymbol{s'}$: $b_1\, b_2 \cdots b_n$

That is, all sites in both sequences are randomly generated, all sites independent of each other.

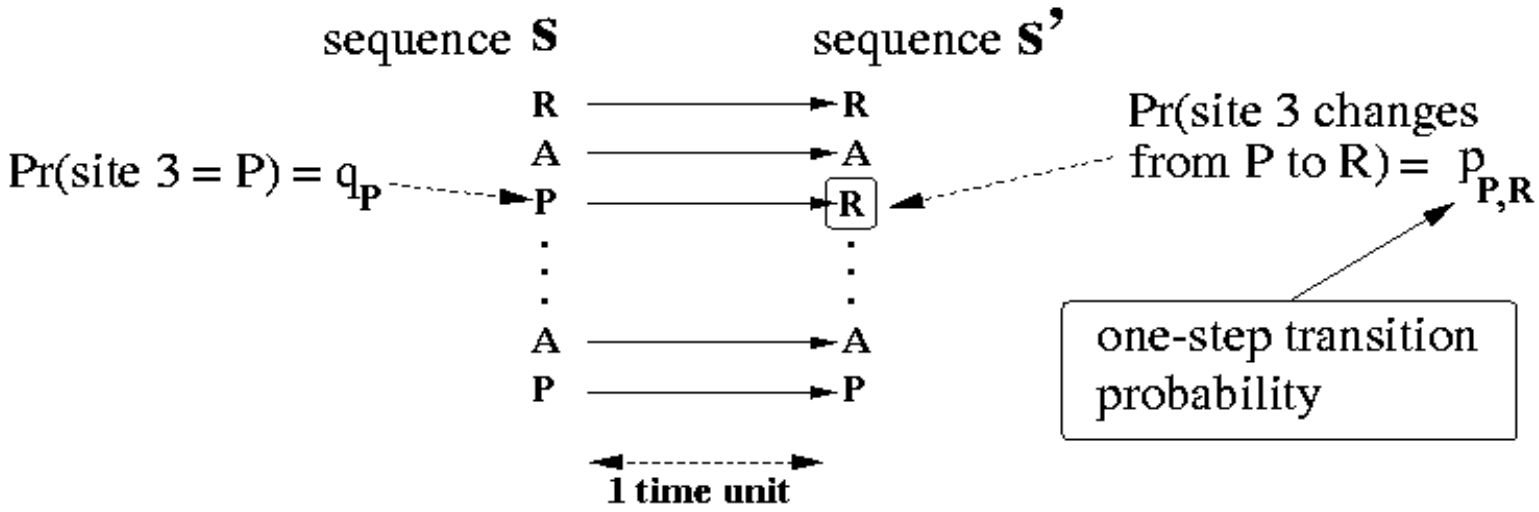
Amino acid j appears with probability q_j.

The probability for getting this *chance* alignment is equal to

$$\mathbf{P}_{H_0}(\textit{the alignment}) = \Big(\prod_{i=1}^{n} q_{a_i}\Big) \cdot \Big(\prod_{i=1}^{n} q_{b_i}\Big)$$

$$= \prod_{i=1}^{n} (q_{a_i} \cdot q_{b_i}).$$

Under H_A, the sites in the sequences are dependent, according to the Markov model described earlier.



Example: $\mathbf{P}_{H_A}(\textit{align } P \textit{ and } R \textit{ in a given site}) = q_P \cdot p_{P,R}$.

Since the different sites evolve independently of each other, we get

$$\mathbf{P}_{H_A}(\textit{the alignment}) = \prod_{i=1}^{n}(q_{a_i} \cdot p_{a_i,b_i}).$$

In principle, we want our score to reflect the 'chance' (or the *odds*) that with $\boldsymbol{s}$ and $\boldsymbol{s'}$ we have aligned evolutionarily related sequences (i.e. basically we want a high score if the odds are high that we have aligned related sequences).

A natural choice for the score is then a comparison of the probabilities under H_A and H_0, respectively:

The **likelihood ratio**:

$$\begin{aligned}\text{alignment score} &= \frac{\mathbf{P}_{H_A}(\textit{the alignment})}{\mathbf{P}_{H_0}(\textit{the alignment})} \\ &= \frac{\prod_{i=1}^{n}(q_{a_i} \cdot p_{a_i,b_i})}{\prod_{i=1}^{n}(q_{a_i} \cdot q_{b_i})} \\ &= \prod_{i=1}^{n} \frac{q_{a_i} \cdot p_{a_i,b_i}}{q_{a_i} \cdot q_{b_i}} = \prod_{i=1}^{n} \frac{p_{a_i,b_i}}{q_{b_i}}.\end{aligned}$$

Or, equivalently, but better for theoretical reasons, one can use the **log likelihood ratio** (Dayhoff et al.: "the **log odds ratio**"):

$$\begin{aligned}\text{alignment score} &= \log\Big(\frac{\mathbf{P}_{H_A}(\textit{the alignment})}{\mathbf{P}_{H_0}(\textit{the alignment})}\Big)\\ &= \log\Big(\prod_{i=1}^{n}\frac{p_{a_i,b_i}}{q_{b_i}}\Big)\\ &= \sum_{i=1}^{n}\log\Big(\frac{p_{a_i,b_i}}{q_{b_i}}\Big).\end{aligned}$$

The entry (a, b) in the **PAM substitution matrix** is then of the form

$$S_{a,b} = \log\Big(\frac{p_{a,b}}{q_b}\Big)$$

(or rounded to the nearest integer for convenience).

Due to the logarithm, we have obtained an *additive* scoring system in a natural way:

alignment:

$\boldsymbol{s}$: $a_1 a_2 \cdots a_n$

$\boldsymbol{s'}$: $b_1\, b_2 \cdots b_n$

Total score: $S(alignment) = \sum_{i=1}^{n} S_{a_i,b_i}$.

***

Adding the scores for each position is equivalent to *multiplying the probabilities* (due to the logarithm)!

$$S(alignment) = \log\left(\frac{\mathbf{P}_{H_A}(the\ alignment)}{\mathbf{P}_{H_0}(the\ alignment)}\right)$$

$$= \log\left(\frac{(q_{a_1} \cdot p_{a_1,b_1}) \cdot (q_{a_2} \cdot p_{a_2,b_2}) \cdots (q_{a_n} \cdot p_{a_n,b_n})}{(q_{a_1} \cdot q_{b_1}) \cdot (q_{a_2} \cdot q_{b_2}) \cdots (q_{a_n} \cdot q_{b_n})}\right)$$

$$= \sum_{i=1}^{n} \log\left(\frac{p_{a_i,b_i}}{q_{b_i}}\right) = \sum_{i=1}^{n} S_{a_i,b_i}.$$

Moreover,

$$S_{a,b} = \log\left(\frac{p_{a,b}}{q_b}\right) < 0$$

if

$$\frac{p_{a,b}}{q_b} < 1 \iff \frac{q_a \cdot p_{a,b}}{q_a \cdot q_b} < 1 \iff q_a \cdot p_{a,b} < q_a \cdot q_b$$

(*i.e.* $S_{a,b} < 0$ *if it is more likely to see* a *and* b *aligned against each other in a random alignment than to see* a *and* b *aligned in a comparison of two related sequences (at 1 PAM distance)*).

Otherwise,

$$S_{a,b} = \log\left(\frac{p_{a,b}}{q_b}\right) \geq 0.$$

PAMn substitution matrix?

For sequences having an evolutionary distance of n PAM units.

Careful: "n PAM units" does not mean that we expect n% of the amino acids to differ... because substitutions can occur at the same site many times!!

Let P be the 1 PAM transition matrix. As always with Markov chains: the n-step transition probabilities $p_{a,b}^{(n)}$ are given as the entries in

$$P^n.$$

The scores are

$$S_{a,b}^{(n)} = \log\left(\frac{p_{a,b}^{(n)}}{q_b}\right).$$

The BLOSUM family of substitution matrices

(Henikoff and Henikoff, 1992).

It requires the use of

- log-likelihood ratios

  (for getting a scoring matrix from a matrix of estimated substitution probabilities)

"BLOSUM" = **BLO**cks **SU**bstitution **M**atrices.

***

Again the scores will be logarithms of likelihood ratios, but this time there are no evolutionary models, and hence no Markov chains and no trees involved.

The likelihoods are obtained by the *statistical analysis of "blocks of aligned sequences"*.

Blocks of aligned sequences

The "blocks" needed to derive the BLOSUMs stem from an *ungapped* multiple alignment of a relatively highly conserved region of a family of proteins (H&H developed a program called "Protomat" for obtaining these blocks).

A difference to the construction of PAM matrices: the kind of data that was used!

- H&H's data was far more extensive: several hundred groups of proteins, at least 2369 occurances of *any particular substitution*.
- difference in concept: Dayhoff et al. used data from closely related proteins and 'extrapolated' (PAM1 $\rightsquigarrow$ PAMn), H&H directly used protein sequences regardless of their evolutionary distances.

Four sample blocks from H&H's Blocks database:

```
WWYIR  CASILRKIYIYGPV  GVSRLRTAYGGRK  NRG
WFYVR  CASILRHLYHRSPA  GVGSITKIYGGRK  RNG
WYYVR  AAAVARHIYLRKTV  GVGRLRKVHGSTK  NRG
WYFIR  AASICRHLYIRSPA  GIGSFEKIYGGRR  RRG
WYYTR  AASIARKIYLRQGI  GVGGFQKIYGGRQ  RNG
WFYKR  AASVARHIYMRKQV  GVGKLNKLYGGAK  SRG
WFYKR  AASVARHIYMRKQV  GVGKLNKLYGGAK  SRG
WYYVR  TASIARRLYVRSPT  GVDALRLVYGGSK  RRG
WYYVR  TASVARRLYIRSPT  GVGALRRVYGGNK  RRG
WFYTR  AASTARHLYLRGGA  GVGSMTKIYGGRQ  RNG
WFYTR  AASTARHLYLRGGA  GVGSMTKIYGGRQ  RNG
WWYVR  AAALLRRVYIDGPV  GVNSLRTHYGGKK  DRG
```

So how are the likelihoods obtained?

Count

- the proportion p_a of times that the AA a occurs somewhere in any block;
- the proportion p_{ab} of times that the AA pair a, b (not necessarily distinct) occurs in the same column of any block (note: there are $20 + \binom{20}{2} = 210$ pairs of amino acids, and a total of $N \cdot \binom{m}{2}$ pairs that have to be taken into account in all the blocks, where N is the number of columns in all blocks together and m is the number of rows in each block, if we assume that this number is the same for all blocks).

The likelihood ratios

p_{ab} is the likelihood / estimated probability for the substitution $a \leftrightarrow b$ under the assumption that the sequences are related.

$p_a p_b$ is the likelihood / estimated probability for the substitution $a \to b$ (also for $b \to a$) under the assumption that the sequences are not related; thus, for $a \neq b$ the likelihood for the substitution $a \leftrightarrow b$ is $2\, p_a p_b$.

A scoring matrix is obtained by the same considerations as for the PAM matrix. Set the entry at position (a, b) as

$$S_{a,b} := \begin{cases} 2 \log_2\left(\frac{p_{ab}}{2\, p_a p_b}\right) & \text{if } a \neq b \\ 2 \log_2\left(\frac{p_{ab}}{p_a p_b}\right) & \text{if } a = b. \end{cases}$$

($2 \log_2$ was what H&H used; using log would give us essentially the same score)

This is not yet the BLOSUM we are looking for!!

The circularity problem. Solution: iteration.

To obtain the initial blocks, a multiple alignment was done
$\longrightarrow$ a substitution matrix is needed for that!!

Henikoff and Henikoff used a simple "unit matrix" for this first alignment (1 for a match, 0 for a mismatch). Then, with the first "BLOSUM matrix" they obtained by the above procedure, they constructed a second set of blocks and a second "BLOSUM matrix". Then, with this second matrix a third matrix was constructed. Only this is the matrix that is recommended to be used.

This gives a **BLOSUM100 matrix**, provided that we have eliminated all identical copies of sequences from the original blocks.

The BLOSUM100 is not very useful!!

The problem of overcounting. Solution: clustering.

The problem of overcounting is solved by clustering those sequences in each block that are "sufficiently close" (i.e. we 'combine' them in a skillful way and regard them as a single sequence).

The result is called a **BLOSUMx (matrix)**, where the number x determines what we mean by "sufficiently close":
We cluster sequences in a block that have x% identity or more.

An "average BLOSUM" often used is BLOSUM62 (is e.g. the default matrix, when you do a BLAST search on NCBI)!

Brief comparison of PAM matrices and BLOSUMs

The circularity problem:

PAM: Not addressed. Simple initial alignment.

BLOSUM: Three iterations of procedure.

The overcounting problem:

PAM: inferring of phylogenetic trees for each block. Substitutions are only counted along the edges of the trees.

BLOSUM: Clustering by the "x% rule" in each block.

The evolutionary distance problem:

PAM: Markov chain theory: different distances are accounted for by n-step transition matrices for different n (higher distances $\widehat{=}$ higher n).

BLOSUM: Clustering by the "x% rule" in each block (higher distances $\widehat{=}$ more clustering $\widehat{=}$ lower x).

Note the last point: The numbers n and x in PAMn and BLOSUMx play opposite roles.

Higher values of n and *lower* values of x both correspond to longer evolutionary distances!!

The n counts "the steps" in the Markov chain used for the evolutionary model.

The x tells up to which percentage of similarity between two sequences in a block the sequences are seen as different.

PAM vs. BLOSUM: which one is better?

Advantages of BLOSUM:

- Simpler model;
- Observation-based, more or less independent of other models and concepts (no Markov chain assumption, trees, maximum parsimony needed);
- Tests suggest that BLOSUM matrices generally are superior to PAM matrices for detecting biological relationships (even if same amounts of data are used).

Advantages of PAM:

- Yields an explicit evolutionary model as a by-product;
- Helps better understanding biological relations.