- The 20 amino acids are quite different.
- Some are big some are small.
- Some are polar others are not.
- Some are hydrophilic others are hydrophobic.
- etc.

- Similar amino acids are more often replaced by each other then dissimilar amino acids.
- Dayhoff et al. 1978: Reverse this relation:
- We measure the similarity of amino acids by observing how often they are replaced by each other.
- Available data: sequence alignments.

Counting pairs of aligned amino acids

- Given a set of reliable pairwise alignments.
- For each pair of amino acids (i, j) we can count how often we observe amino acid i in the first sequence and aligned to it amino acid j in the second sequence.

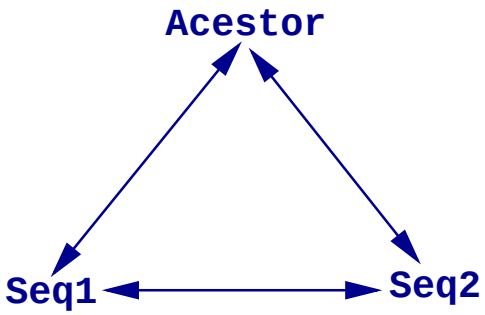
```
 Comparison of:
(A) mariner.seq >A26491    probable transposition protein              - 345 aa
(B) tc1.seq     >TC1       P03934 273AA                               - 273 aa
 using matrix file: BLOSUM50, gap penalties: -14/-4

  24.7\% identity in 97 aa overlap; score:  109


A26491 IFLHDNAPSHTARAVRDTLETLNWEVLPHAAYSPDLAPSDYHLFASMGHALAEQRFDSYESVKKWLDEWFAAKDDEFYWRGIHKLPER
       .: .:: :.:::.  ::. ..    ..:   . :::: : . ::.  . . :.  : .. ..  . :. . :       . : ..:.:
TC1    VFQQDNDPKHTSLHVRSWFDRRFVDLLDWPSQSPDLNPIE-HLWEELERRLGGIRASNADAKFNQLPNAWKAIPMSVIHKLIDSMPRR
```

$$\#(D, E) = 4$$

$$\#(N, F) = 1$$

However, we can observe the process only indirectly by compairing descendants.

For a time reversible model this is no problem:



The differences between Seq1 and Seq2 can be modelled by a single time reversible model.

Symmetry of the observations

- Deciding which of the sequences is the first and which is the second sequence in an alignment is completely arbitrary. Hence, we should not distinguish between observing i in the first and j in the second or j in the first and i in the second sequence.
- For example in
  MLKEVAKSHH
  MKHEVKHSKH
  we count the (H, K) pair 3 times.
- We can summarize the relative pair frequencies

$$m_{ij} = \frac{\#\text{positions where } i \text{ is aligned to } j}{\Sigma \text{ Length of alignment}}$$

  in a 20 by 20 matrix M_{emp}.
- Due to the symmetry of the observations, M_{emp} is symmetrical too.

Dayhoff's calculations

- Derive transition probabilities from M_{emp}.
- Following the original paper we treat mismatch and match observations separately.
  Mismatches first: assume $i \neq j$:

$$P_{ij} = P[i \text{ mutates}]P[i \to j \mid i \text{ mutates}].$$

- We want to estimate the term on the left, we have data for both terms on the right.

Mutability

- First calculate $P[i \text{ mutates}]$ the mutability of the amino acid i.
- This term can be estimated by

$$m_i = \frac{\sum_j M_{ij}}{\sum_{j,k,k \neq j} M_{kj}}.$$

- $P[i \to j \mid i \text{ mutates}]$ can be estimated by

$$\frac{M_{ij}}{\sum_{k \neq i} M_{ik}}$$

- The diagonal entries of P are consequently

$$P_{ii} = 1 - m_i$$

.

Questions

- Which alignments should be used?
- To which time point t do our observations belong?
- What is the unit of time?
- What is a good time point for deriving a model based score function for protein alignment?

Calibration and PAM Distance

- The time point $t = 1$ corresponds to 1% expected mismatch positions in the observed alignments.

$$P[X_t \neq X_{t+1}] = 0.01$$

- This unit of time is called 1 PAM
  "Point Accepted Mutations"
- 2 PAM correspond to the effect the Markov chain has, if it runs twice as long. In general this results in less than 2% expected mismatch positions, since with some small but positive probability one of the already changed positions mutates a second time.

Dayhoff's data

- Dayhoff et al. only used closely related alignments in the range of 0 to 17 PAM.
- They treated all this data in the same way. Hence they ignored the small differences in the degree of divergence.
- Having M_{emp}, they calculated transition matrices $P(t)$ as described above ... this also gives a rate matrix Q and a stationary distribution π.
- The stationary distribution π reflects the relative frequency of amino acids in the data.

Symmetry and time reversibility

- Since, M_{emp} is symmetrical, the resulting Markov chain is always time reversible
- This is another argument in favor of time reversible models.
- Even if evolution is not a reversible process, we do not have observations that would allow us to distinguish between directions.

Calibration continued

- The expected number of mismatch positions for $t = 0$ is 0. It is then continuously growing with t.
- Hence, there must be a t that corresponds to 1 PAM.
- This time point can be calculated efficiently by diagonalisation of the transition matrices $P(t)$.
- Dayhoff et al. made use of the linear approximation

$$P(t) = I + tQ$$

for small t and calibrated by transforming the mutabilities:

$$m_i \to m_i/100\pi_i.$$

A problem:

- Sequences that are 1 PAM apart are very similar, alignment is usually unambiguous and can essentially be done by hand.
- In real alignment problems, we are dealing with sequences that are fare more remote.
- For the challenging alignment problems the models used to build score matrices, should reflect pair frequencies in distantly related sequences.

Extrapolation

Dayhoff et al., having a lot of faith in their model, suggest:

- Use the 1 PAM transition matrix P. (A little bit of evolution)
- Calculate the corresponding 250-step transition matrix P^{250}. (A lot of evolution)
- Calculate the corresponding joint distribution of sequences that are 250 time units (PAMs) apart.

$$m(250)_{ij} = P_{ij}^{250}\,\pi_i$$

The PAM family of score matrices

- We can calculate the famous PAM250 Score matrix just by

$$PAM(250)_{ij} = 10 \log_{10} \left(\frac{m(250)_{ij}}{\pi_i \, \pi_j} \right)$$

- Actually, we can extrapolate a score matrix for any PAM distance by

$$PAM(t)_{ij} = 10 \log_{10} \left(\frac{m(t)_{ij}}{\pi_i \, \pi_j} \right)$$

- Dayhoff et al. have suggested PAM(250), today PAM(160) is assumed to be a better choice.

Improvements

- The PAM matrices were derived in 1978 from a relatively small number of alignments. Today we have much much more data.
- The PAM matrices are estimated from observations of only very closely related sequences. A position that mutates that early is a fast evolving position. When aligning remote sequence pairs we are especially interested in aligning conserved regions correctly. These might follow different models.
- It is desirable to fit models using more data including more distantly related sequences

We discuss two approaches

- The BLOSUM matrices
- The variable time matrices VT

BLOSUM

- Derived by Steven Henikoff and Jorja Henikoff 1992
- Idea:
  Forget about the Markov model, but select your data carefully.
- Blocks database: contains conserved ungapped segments from protein families.
- A block is a short ungapped interval in a multiple alignment of proteins.
- The BLOSUM score matrix is derived from these multiple alignments.

From Blocks to BLOSUM

- Given a set of blocks:
- Consider all pairs of positions in this set of multiple alignments. (Compare sum of pairs score)

  YVHKL

  YVYKL

  MVKKL

  The first column results in the pairs (Y, Y) and (Y, M) counted twice.
- For each pair of amino acids (i, j) count its occurrences.
- Normalize by the frequencies of i and j in the blocks. (Quick and dirty approach)
- How can we focus on a certain degree of divergence?

- Fix a percentage identity x between 50% and 80%.
- Remove rows from the blocks such that the remaining rows all have less then x% pairwise identity.
- Count pair frequencies $m(x)_{ij}$ in these blocks and normalize them.
- $(f_1, \ldots f_{20})$ are the relative frequencies of the amino acids in the reduced blocks.
- We get the Score matrices:

$$\texttt{BLOSUM}(x)_{ij} = 2 \log_2 \left(\frac{m(x)_{ij}}{f_i \; f_j} \right).$$

- The BLOSUM matrices are based on observations from remote sequences.
- They are derived from multiple alignments instead of pairwise alignments. Multiple alignments are in general more reliable.
- In most applications, especially database searches, the BLOSUM matrices proved to be better than the PAM matrices.
- BLOSUM62 is the most widely used scoring matrix today.
- But, they are not based on a model of evolution.

- Is there a possibility to have both? A good score matrix based on a large set of observations including divergent sequences and a corresponding model of evolution.
- What would be the problem if we just applied Dayhoff's method to this kind of data?

```
        10        20        30        40        50        60
VCKITPHSSNKSYPDGVYGTSGSANDDKQDAPHYIGTLDMTAFGSLFHEDDFELNFGTAK ...
::::::::. .::.:::::::: :::: .:::::.:::::::::::::::::::.:::.::::.:
VCKITPHAPHKSHPDGVYGTPGSANADRQDAPNYIGTLDMTAFGSLFHEDEFELTFGTTK ...
        10        20        30        40        50        60
```

$$\#(D, E) = 1$$

$$\#(N, F) = 0$$

```
        10        20        30        40        50        60
KLNELIPTRLDRKGLQSGGKVDRYQDEKYRKVGSPYFKKSHARKLAGSLTSDAITTLVRA ...
....:   ::.: ::::::::. :: . .    :  :   :..: : :. :.::   ::..
RVSDLYGIRLERAGLQSGGKLARYVEASLTTHGLAYNMASRTRLLQGAHTGDASDGLVKT ...
        10        20        30        40        50        60
```

$$\#(D, E) = 3$$

$$\#(N, F) = 1$$

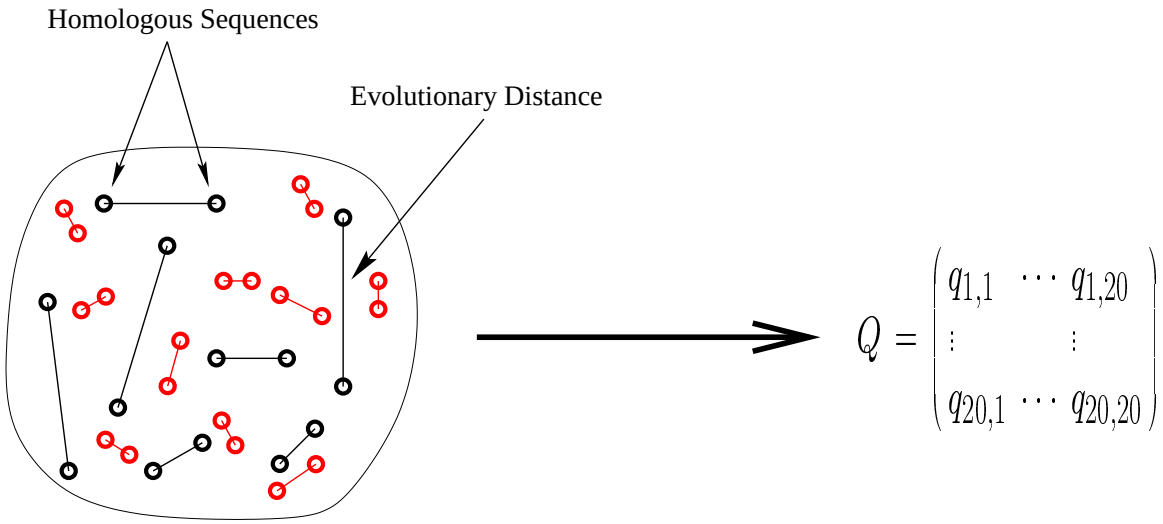
```
     10        20        30        40        50        60
PKNDSHTQVKEGTEQTFVLPKAHAASKLVEDLLGAGVDSKPNGAYTQESDPSSVPEGVTD ...
:. .. :  :.:.  . :  . ::.  ..  . : .:..:  :   ...::  . .:. .
PQFEGFTTGKDGAPLAAVQKQYHATVMFIVMMGGFAVEQKGFGFRGSDKDPCHTSHGLLE ...
     220       230       240       250       260       270
```

$$\#(D, E) = 5$$

$$\#(N, F) = 2$$

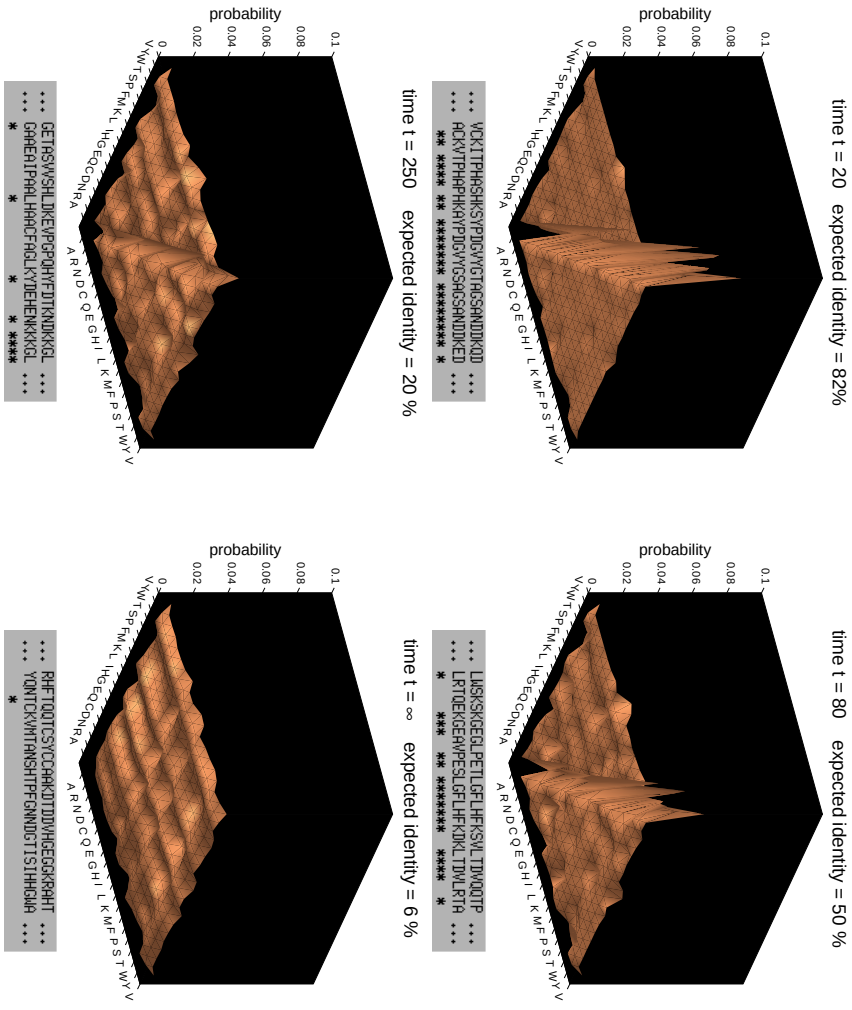
The problem is:

- Observations from closely related sequences correspond to a different model, than observations from distantly related sequences.
- On the other hand, a model for closely related sequences implies a model for distantly related sequences and vice versa.
- If we fit separate models for both types of alignments, we run into inconsistencies.
- How can we estimate a single model consistently?



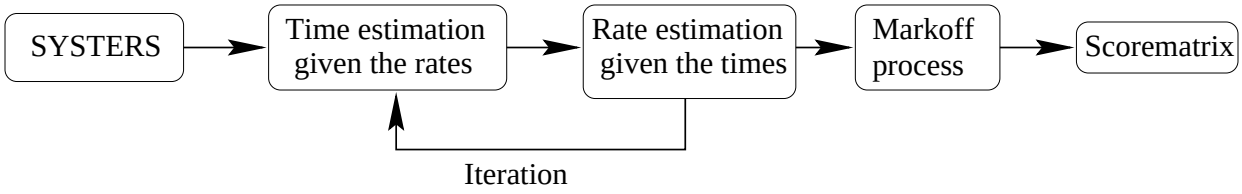
Input Data

Rates


time t = 20 expected identity = 82%
time t = 80 expected identity = 50 %
time t = 250 expected identity = 20 %
time t = ∞ expected identity = 6 %
probability
0
0.02
0.04
0.06
0.08
0.1
A R N D C Q E G H I L K M F P S T W Y V
V V W T S P F M K L I H G E Q C D N R A
*** VCKITPHASHKSYPDGVYGTAGSANDDKQD ***
*** ACKVTPHAPHKAYPDGVYGSAGSANDDKED ***
** **** ** ******* ******** * ***
*** GETASVWSHLDKEVPGQHYFDTKNDKKGL ***
*** GAAEAIPAALHAACFAGLKYDEHNKKKGL ***
* * * * **** ***
*** LWSKSKGEGLPETLGFLHFKSVLTDVQQTP ***
*** LRTQEKGEAVPESLGFLHFKDKLTDVLRTA ***
* *** ** ******* **** * ***
*** RHFTGQTCSYCCHAKGTDDVHGEGGKRAIT ***
*** YQNTCKVMTANSHTPFGNNDGTISIHHGWA ***
*

- We will base the estimation on pairwise alignment data, as in the original Dayhoff model.
- A priori, we have no clue what the correct model of evolution is, nor do we have any idea what the degree of divergence of the individual sequence pair is.
- It is clear, that we need to have information on the degree of divergence (the time interval in which the Markov chain is operating), if we want to estimate a model. (Model estimation)
- On the other hand, a model is perfectly suited for estimating these numbers.(Time estimation)

- Solution:
  We start with a known model (e.g. Dayhoff's model) and then iterate through several rounds of time estimations and model estimations.


SYSTERS
Time estimation given the rates
Rate estimation given the times
Markoff process
Scorematrix
Iteration

- Assume we have all the necessary information on the time of divergence (T) for all the alignment data (A).
- How can we estimate a model of protein evolution from time inhomogenous alignment data?
- We will discuss:
  - Maximum Likelihood via rate matrix.
  - Integral estimation via resolvent.

Maximum Likelihood

- The pair (Q, π) specifies the model completely.
- Choose (Q, π) such that the likelihood of the given information (A, T) is optimal.

$$
\begin{aligned}
(\widehat{\pi}, \widehat{Q}) &= \underset{\pi, Q}{\operatorname{argmax}} \, \mathcal{L}(\pi, Q | T, A) \qquad (1) \\
&= \underset{\pi, Q}{\operatorname{argmax}} \sum_{i,j} N_{ij}^{(k)} \log((F e^{t^{(k)} Q})_{ij}),
\end{aligned}
$$

  where $N_{ij}(k)$ counts aligned amino acid pairs in alignments of divergence $t^{(k)}$, F is a diagonal matrix with entries π_i and Q is a rate matrix.

- The parameterization of (Q, π) must ensure that we end up with a time reversible and calibrated model.

Problem

- The maximum likelihood method can deal with time divergent observations.
- However, calculating the maximum is computationally demanding. Only relative small amounts of input data can be handled.
- Much more data is available.
- Hence, we need a more efficient procedure.

Problem

- Why is Maximum Likelihood slow ?
- Whenever we are evaluating the likelihood of a candidate rate matrix Q, we need to calculate $\exp(t^{(k)}Q)$.
- This requires a diagonalisation of Q.

The resolvent

- For $\alpha > 0$, we define a weighted time average of $P(t)$:

$$R_\alpha = \int_0^\infty e^{-\alpha t} P(t)\, dt.$$

- R_α is called a resolvent of $P(t)$.
- The resolvent is related to the rate matrix by

$$\alpha I - R(\alpha)^{-1} = Q \quad \text{for all} \quad \alpha > 0.$$

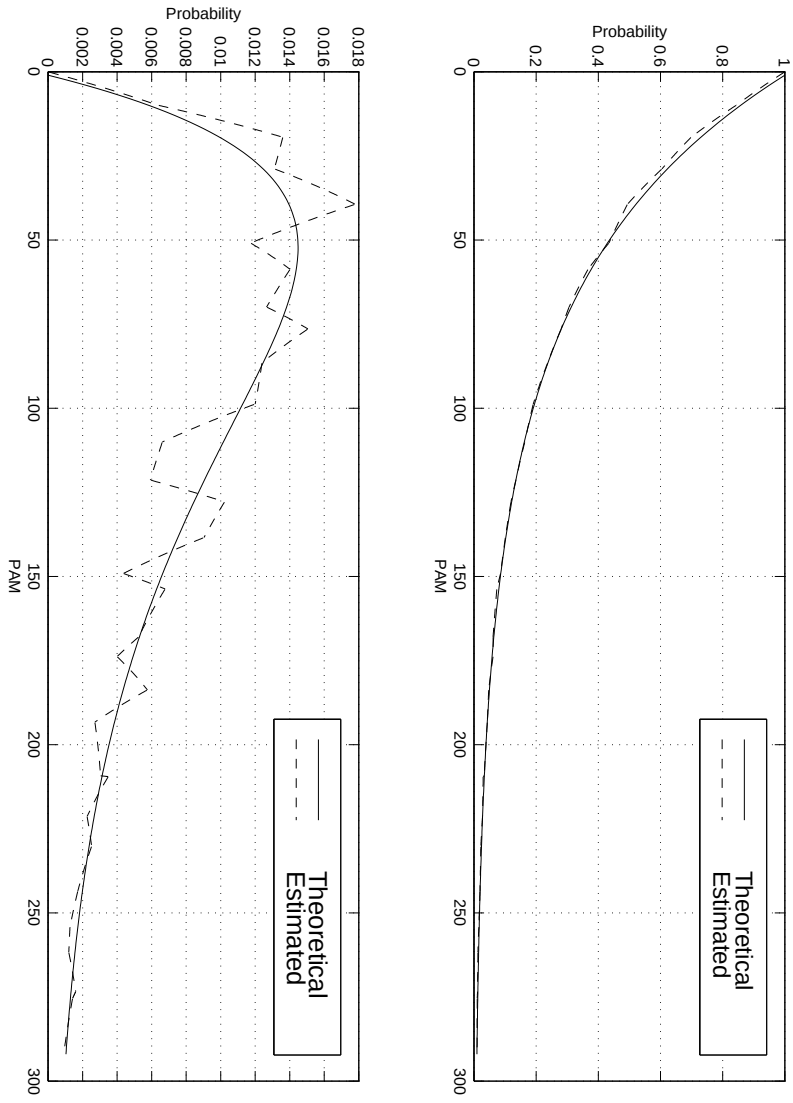
- Idea: Estimate the integral

$$R(\alpha) = \int_0^\infty e^{-\alpha t} P(t)\, dt.$$

- $R_{ij}(\alpha)$ can be estimated from $P_{ij}(t)$ independently from the other entries in R and P.

- Since $P_{ij}(t)$ is a continuous function in t, we only need estimators of $P_{ij}(t)$ on some sufficiently dense set of time points $t_1, \ldots t_n$.

- Due to the weights $e^{-\alpha t}$, high values of t have little influence on the integral.
- In fact we can choose α such that our observations coincide with the most important region for the integral.
- We calculate the integral by linear interpolation of the time specific estimates.


Probability
PAM
Theoretical
Estimated
Probability
PAM
Theoretical
Estimated

- We have discussed the problem of fitting a model to alignment data, if the degree of divergence (time, distance) of all pairs of sequences is known.
- What remains is the complementary problem of estimating the degree of divergence, if a complete model is given.
- We discuss:
  - Maximum Likelihood
  - The log-det-formula

Maximum Likelihood

- By definition, the maximum likelihood estimator $\hat{t}$ is the time t that maximizes the likelihood

$$\mathcal{L}(t|A, Q, \pi).$$

- We have

$$\begin{aligned} 0 &= \frac{d}{dt}\mathcal{L}(t|A, Q, \pi) \\ &= \sum_{ij} N_{ij} \frac{d}{dt} \log(Fe^{tQ})_{ij} \end{aligned}$$

- Using the forward-backward equation, the estimated time of divergence is the solution of

$$\sum_{ij} N_{ij} \frac{(P(t)Q)_{ij}}{P(t)_{ij}} = 0.$$

- The equation can be solved numerically.

The log-det formula

- Let $(\lambda_1, \ldots \lambda_{20})$ be the eigen values of the rate matrix Q, and let $D(t)$ be a diagonal matrix with entries $(e^{t\lambda_1}, \ldots, e^{t\lambda_{20}})$.
- Diagonalisation of $P(t)$ yields

$$\begin{aligned} \log(\det(P(t))) &= \log(\det(S\, D(t)\, S^{-1})) \\ &= \log(\det(S)\, \det(S^{-1})\, \det(D(t))) \\ &= \log\left(\Pi_i\, e^{t\lambda_i}\right) \\ &= t \sum_i \lambda_i. \end{aligned}$$

- Hence,

$$\frac{\log(\det(P(t)))}{\log(\det(P(1)))} = t.$$

The log-det formula continued

- We have,

$$\frac{\log(\det(P(t)))}{\log(\det(P(1)))} = t.$$

- Since P is given, we can calculate the normalizing constant $\log(\det(P(1)))$.
- t is unknown, but we can estimate $P(t)_{ij}$ by

$$P_{\mathsf{emp}} = (M_{\mathsf{emp}})_{ij} / f_i,$$

where $(M_{\mathsf{emp}})_{ij}$ is the relative frequency of the the pair (i,j) and f_i is the relative frequency of amino acid i.

The log-det formula continued

- In total, this gives us an estimator for t:

$$\hat{t} = \frac{\log(\det(P_{\mathtt{emp}}))}{\log(\det(P(1)))}$$

- Note, that

$$\log(\det(P_{\mathtt{emp}})$$

  is proportional to t and does not depend on the real model at all.

The variable time matrix VT160

- Mueller and Vingron 2000
- The VT-matrices are based on large set of input alignments from the SYSTERS database.
- It is calculated by iterative updates of model and time estimates.
- Time estimation is done by Maximum Likelihood.
- Models are derived using the resolvent.
- The number 160 refers to 160 PAM
- The matrix is quite similar to BLOSUM62.
- Different to BLOSUM it is based on a complete stochastic model.