doi:10.1006/jmbi.2000.4061 available online at http://www.idealibrary.com on IDEAL® J. Mol. Biol. (2000) **302**, 265–279

JMB

Statistical Alignment: Computational Properties, Homology Testing and Goodness-of-Fit

J. Hein[1]*, C. Wiuf[2], B. Knudsen[1], M. B. Møller[3] and G. Wibling[3]

[1]*Department of Genetics and Ecology The Institute of Biological Science, University of Aarhus, Building 540, Ny Munkegade, 8000 Århus C Denmark*

[2]*Department of Statistics University of Oxford, 1 South Parks Road, Oxford, OX1 3TG UK*

[3]*The Institute of Computer Science, University of Aarhus Building 550, Ny Munkegade 8000 Århus C, Denmark*

**Corresponding author*

The model of insertions and deletions in biological sequences, first formulated by Thorne, Kishino, and Felsenstein in 1991 (the TKF91 model), provides a basis for performing alignment within a statistical framework. Here we investigate this model.

Firstly, we show how to accelerate the statistical alignment algorithms several orders of magnitude. The main innovations are to confine likelihood calculations to a band close to the similarity based alignment, to get good initial guesses of the evolutionary parameters and to apply an efficient numerical optimisation algorithm for finding the maximum likelihood estimate. In addition, the recursions originally presented by Thorne, Kishino and Felsenstein can be simplified. Two proteins, about 1500 amino acids long, can be analysed with this method in less than five seconds on a fast desktop computer, which makes this method practical for actual data analysis.

Secondly, we propose a new homology test based on this model, where homology means that an ancestor to a sequence pair can be found finitely far back in time. This test has statistical advantages relative to the traditional shuffle test for proteins.

Finally, we describe a goodness-of-fit test, that allows testing the proposed insertion-deletion (indel) process inherent to this model and find that real sequences (here globins) probably experience indels longer than one, contrary to what is assumed by the model.

Keywords: statistical alignment; homology testing; goodness-of-fit

Introduction

Statistically well founded methods have become increasingly used over the last decade in the analysis of biological sequences. This has not been the case in the alignment of sequences, partly because of the general conception that the statistical approach to alignment is computationally too slow and partly due to the lack of user-friendly programs. Often, the sequences are aligned using parsimony or similarity based methods (optimisation alignments). The alignment is subsequently treated as a series of columns that are independent realizations of a substitution process on a phylogeny that is to be estimated. It is an inconsistent approach to first use parsimony/similarity and then halfway in the analysis switch to a statistical approach. In addition, the alignment created by parsimony/similarity can create unknown biases in the estimation of substitutional parameters, as these procedures will align to create as much identity within each column as possible.

The first attempt to do statistical alignment was done by Bishop & Thompson (1986), with approximate likelihood calculations. Thorne *et al.* (1991) introduced an exact method (TKF91). In this framework, there will not be one alignment, but all possible alignments will contribute to the likelihood of the two observed sequences. Should one alignment be highlighted, it could be the alignment that contributed the most to the likelihood. Alignments can have runs of gap signs, which in a parsimony/similarity setting would be interpreted as a longer insertion/deletion (indel), but here it would be the consequence of several neighbouring indels of single nucleotides. Most optimisation alignment methods interpret runs of gap signs as a single event, even when they may be the result of multiple independent insertions and deletions. Nevertheless, indels longer than one nucleotide or amino acid must occur biologically and the TKF91 model does not allow for that. Thorne *et al.* (1992) tried to incorporate this in the model by letting insertions

E-mail address of the corresponding author: jotun.hein@biology.au.dk

and deletions involve fragments that each had a geometrical distribution. For computational reasons each fragment would have to be treated as an unbreakable unit, which is not a biologically well founded assumption.

An alternative approach to statistical alignment has been taken by Allison & Wallace (1994), Zhu *et al*. (1998), and Mitchison (1999). These are Bayesian approaches and also differ from the TKF91 model in not being based on an evolutionary process, but on a probability measure on alignments directly.

Pairwise sequence alignment, homology testing and multiple alignment has great importance in sequence analysis and there is an increasing awareness of the advantages of statistical approaches within the bioinformatics community. Therefore, statistical alignment and its ramifications deserve to be pursued with much greater intensity.

Theory

The TKF91 statistical model of DNA evolution is a continuous time model with a state space consisting of all sequences over an alphabet of nucleotides (yielding DNA sequences) or amino acids (yielding proteins) that includes the empty sequence. If we, for any possible sequence, can define the waiting time to the first event (insertion, deletion or change of single element) to occur and the probability of all the possible events, the model has been characterised.

Modelling substitutions

Since the novelty of the TKF91 model is in the modelling of the indel process, the substitution process will receive little attention here. Almost all substitution models are continuous time Markov models on the state space of nucleotides or amino acids. To define such a model, the rate matrix, Q, must be specified. This matrix is 4 by 4 for nucleotides and 20 by 20 for amino acids. Off diagonal elements are non-negative and the sum of each row is zero. This matrix describes the intensity of different substitution events over an infinitesimal time period. The transition probabilities over a longer time interval, t, can be obtained by:

$$P(t) = e^{tQ} = \sum_{k \geqslant 0} \frac{(tQ)^k}{k!}.$$

$P_{i,j}(t)$ refers to the probability that i has changed to j after a time t. Normally, it is also assumed that the process is time reversible, i.e. that $\pi_i P_{i,j}(t) = \pi_j P_{j,i}(t)$, where the π_i terms are the equilibrium frequencies of the nucleotides/amino acids in the process. In this case, the evolutionary process can be viewed from any time perspective, the total probabilities involved will be the same. This has computational advantages, since the evolutionary history can be rooted anywhere.

Since the time, t, and the rate matrix, Q, always appear together as a product in these calculations, it is not possible to estimate them individually. It is often convenient to scale them, so one event is expected per unit time per position in the equilibrium process. This is equivalent to placing the restriction $\Sigma_i \pi_i Q_{ii} = -1$ on the rate matrix.

The main difference between substitution processes on nucleotides and on amino acids, stems from the larger set of amino acids. In principle a 20 by 20 rate matrix with many parameters could be inferred, but there are not such clear distinctions as the transversion/transition bias in the case of nucleotides. This means that more crude ways of choosing a Q matrix, than maximum likelihood estimates of individual entries are used, since there are too many parameters to be estimated. Dayhoff *et al.* (1978) pioneered the use of Q-matrices obtained from counting mutations in comparisons of similar protein sequences. Due to the inability to distinguish where A has mutated to B or *vice versa*, such matrices will be symmetric and give rise to equilibrium distributions where all the amino acids are equally likely. This is in conflict with the frequencies that can be observed in real sequences. Fortunately, it is possible to modify a symmetric matrix, so it will give rise to the observed frequencies of the 20 amino acids. In addition, the resulting matrix will yield a mutation process that is time reversible. The rate matrix used here was obtained from Ziheng Yang (personal communication).

The TKF91 model of the indel process

The basic model

The statistical model of sequence evolution incorporating insertions and deletions can be viewed as a Markov model with all sequences as possible states. The indel part of this model can be illustrated by the use of links connecting the letters (nucleotides or amino acids) of the sequences. Each letter has a mortal link associated to it, on its right. The left end of the sequence has an immortal link. Consider an example, the DNA sequence AGG:

$$\bullet A \star G \star G \star$$

Mortal links are symbolised by ★, while immortal links are symbolised by •. Links can give birth to new links to their right. Along with the birth of such a link, comes a letter drawn from the equilibrium distribution. The mortal links can, as the name suggests, also die. When a link dies, it takes its letter (to the left) with it. The transition in the Markov model, when the mortal link between the two G residues gives birth to a new link and a C, is shown here:

$$\bullet A \star G \star G \star \rightarrow \bullet A \star G \star C \star G \star$$

The new link is the one to the right of the C. If the first mortal link dies, it looks like this:

$$\bullet A \star G \star G \star \rightarrow \bullet G \star G \star$$

This process can be visualised in the following way:

$$\begin{array}{ccccccc} \bullet & A & \star & G & \star & G & \star \\ I_0 & S_1 & I_1/D_1 & S_2 & I_2/D_2 & S_3 & I_3/D_3 \end{array}$$

I, *D* and *S* terms are all independent processes that describe insertions, deletions and substitutions, respectively. For each of these, there will be an exponential waiting time for an event to occur. Whichever fires first, determines the next event.

I_i has intensity parameter λ. When *I* fires first, a nucleotide will be inserted, according to the equilibrium frequency of the substitutional process, with a mortal link on its left side. The newborn nucleotide and associated mortal link will be inserted to the right of the I_i that fired.

D_i has intensity parameter μ. When a D fires first, the link and its nucleotide (to the left) will be removed. The immortal link will never be deleted, since it is not associated with a deletion process. The deletion rate has to be bigger than the insertion rate ($\mu > \lambda$), otherwise the sequence length would grow towards infinity. This process will have a stationary distribution on sequences:

$$P(s) = \gamma_l \pi_A^{\#A} \pi_C^{\#C} \pi_G^{\#G} \pi_T^{\#T},$$

where, for $l \geqslant 0$:

$$\gamma_l = \left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^l.$$

l is the length of the sequence and #*A* (#*C*, #*G*, #*T*) indicates the number of *A* residues (*C*, *G*, *T* residues) in the sequence. This indel process is time reversible, thus, if the substitution process is also time reversible, the process on full length sequences is reversible. This is not a pure birth-death process because of the immortal link, but can be viewed as a birth-death process with immigration.

Evolution over a time period

Above, it was described how a sequence would evolve in an infinitesimal time interval. Thorne *et al.* (1991) also described the transition probabilities for a fixed time interval (Figure 1). Due to the independence between links, it is sufficient to describe what happens to a single link. There is an insertion process associated with the immortal link, with the transition probability $p''_n(t)$, the probability that the immortal link has left itself and $n-1$ mortal descendants after time *t*. There is an indel process associated with the mortal links, with two sets of transition probabilities, depending on whether the link has survived or not. The probability is $p_n(t)$ if the link has survived and left *n* (mortal) descendants, including itself. If the link has not survived, the probability is $p'_n(t)$, again with *n* being equal to the number if descendants. This last distinction between survival and non-survival is necessary, since only in the first case will a nucleotide exist both at time zero and time *t* and the probability of going from one nucleotide to *n* nucleotides will involve a substitutional probability. The surviving children of a nucleotide will be to the right of the parent nucleotide. In the following, the evolutionary process parameters, including *t*, will often be suppressed.

The functions p_k, p'_k and p_k'' are modified geometric distributions. The function describing immortal link (p_k'') is the geometric function shifted so that it starts in one instead of zero. The function describing the case where a mortal link survives (p_k) is again shifted to start in one and every position has been multiplied with the probability of survival ($e^{-\mu t}$). The probability of the nucleotide not surviving is $1 - e^{-\mu t}$. In this case, there will be a probability for having zero surviving offspring ($\mu\beta(t)$), with:

$$\beta(t) = \frac{1 - e^{(\lambda-\mu)t}}{\mu - \lambda e^{(\lambda-\mu)t}}$$

and the remaining distribution (no survival but surviving descendants) has again the same geo-

Time	Immortal	Mortal	Mortal
0	$\bullet$	$\star$	$\star$
t	$\underbrace{\bullet\ \star \cdots \star}_{k}$	$\underbrace{\star\ \star \cdots \star}_{k}$	$-\ \underbrace{\star \cdots \star}_{k}$
Number of offspring, k	$p''_k(t)$	$p_k(t)$	$p'_k(t)$
0	0	0	$\mu\beta(t)$
1	$1 - \lambda\beta(t)$	$e^{-\mu t}(1 - \lambda\beta(t))$	$(1 - e^{-\mu t} - \mu\beta(t))(1 - \lambda\beta(t))$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	$p''_1(t)(\lambda\beta(t))^{n-1}$	$p_1(t)(\lambda\beta(t))^{n-1}$	$p'_1(t)(\lambda\beta(t))^{n-1}$

where $\beta(t) = \frac{1-e^{(\lambda-\mu)t}}{\mu-\lambda e^{(\lambda-\mu)t}}$

Figure 1. The probability distributions of different link configurations after a period *t*. The fate of a mortal link has to be split into two cases to accommodate the possibility of substitutional evolution. Since $\lambda < \mu$, $\beta(t)$ is always smaller than one.

metric tail, but with total mass adjusted to give total probability of one to all possible fates of a mortal link.

These transition probabilities of substitutions and of the fates of mortal and immortal links, allow a dynamic programming algorithm calculating $P(s^{(1)}, s^{(2)})$ to be formulated, where $s^{(1)}$ and $s^{(2)}$ are the two complete sequences.

The probability of the sequences and one specific alignment is easily described in terms of the substitution probabilities and p functions. Regard the following example:

$$\begin{array}{cccccc} \bullet & A & \star & T & \star & - \\ \bullet & C & \star & T & \star & G & \star \end{array}$$

Here, $P(s^{(1)}, s^{(2)}, \text{alignment}) = (p''_1)(\pi_{Ap1}P_{AC})(\pi_T P_{TTp2}\pi_G)$. Each parenthesis describes the probability of a link, the associated nucleotide (in the case of mortal links) and their fates. The first parenthesis is the probability that the immortal link survives with one descendant — itself. The last parenthesis says that a T was chosen, the T evolved into a T, the link associated to the T had two descendants including itself and the extra descendant is G. To calculate the probability of two sequences without conditioning on the alignment, it is necessary to sum over all alignments weighted with their probabilities according to the TKF91 process.

A simpler recursion

In the following, $s_i^{(1)}$ is the prefix of length i in $s^{(1)}$ and $s_j^{(2)}$ is the prefix of length j in $s^{(2)}$. $s^{(k)}\ [i]$ refers to the ith nucleotide in the kth sequence. $\pi_{s^{(k)}[i:j]}$ is the probability of the elements from i to j in sequence k in the equilibrium distribution of the substitution process. The probability of the complete sequences, $P(s^{(1)}, s^{(2)})$, can be written as $P(s^{(1)})P(s^{(2)}|s^{(1)})$. The first factor is straightforward to calculate and we will focus on calculating the second only. The reformulation of the algorithm below represents a simplification and acceleration relative to the original TKF91 formulation. Firstly, we only make a recursion for $P(s^{(2)}|s^{(1)})$, while they incorporated the probability $P(s^{(1)})$. Secondly, we only need one or two quantities per entry (i,j) while they needed three quantities. The resulting recursion (5) is as simple as the most basic optimisation alignment algorithm. The basic recursions are summarised in Table 1.

It is possible to decompose the probability $P(s_j^{(2)}|s_i^{(1)})$ by partitioning the conversion of $s_i^{(1)}$ to $s_j^{(2)}$, into the fate of $s_{i-1}^{(1)}$ and the fate of $s^{(1)}[i]$, since these fates are independent (Figure 2).

This example illustrates why it is necessary to distinguish whether a nucleotide survives or not. Only in the former case has substitutional evolution been observed. If the first sequence is empty, i is zero and the immortal link must have evolved into $s_j^{(2)}$, which has probability $p''_j\pi_{s^{(2)}[1:j]}$.

The above illustration can be summarised in following recursion:

$$\begin{aligned} P(s_j^{(2)}|s_i^{(1)}) = p'_0 P(s_j^{(2)}|s_{i-1}^{(1)}) + \sum_{1 \leqslant k \leqslant j} P(s_{j-k}^{(2)}|s_{i-1}^{(1)}) \\ \times (p_k P_{s^{(2)}[i],s^{(2)}[j-k+1]}\pi_{s^{(2)}[j-k+2:j]} \\ + p'_k\pi_{s^{(2)}[j-k+1:j]}) \end{aligned} \quad (1)$$

$$P(s_j^{(2)}|s_0^{(1)}) = p''_j\pi_{s^{(2)}[1:j]} \quad (2)$$

This recursion allows an $O(l^3)$ (l denoting average sequence length) algorithm to be formulated for calculating $P(s^{(1)}, s^{(2)})$.

Due to the geometric tails of the p functions, this formulation can be changed, resulting in an $O(l^2)$ algorithm. The trick applied here is highly reminiscent of the method used by Gotoh (1982) in reducing the computational complexity of an optimisation alignment algorithm from $O(l^3)$ to $O(l^2)$.

Define $R_{i,j} = P(s_j^{(2)},\ s^{(2)}[j]$ is a descendant of $s^{(1)}[i]|s_i^{(1)})$. This will be the sum on the right side of (1). The first recursion above can now be written as:

$$\begin{aligned} R_{i,j} =& (p_1 P_{s^{(1)}[i],s^{(2)}[j]} + p'_1\pi_{s^{(2)}[j]})P(s_{j-1}^{(2)}|s_{i-1}^{(1)}) \\ &+ \lambda\beta\pi_{s^{(2)}[j]}R_{i,j-1} \end{aligned} \quad (3)$$

$$P(s_j^{(2)}|s_i^{(2)}) = R_{i,j} + p'_0 P(s_j^{(2)}|s_{i-1}^{(1)}) \quad (4)$$

The functions $p_1 P_{s^{(1)}[i],s[j]^{(2)}} + p_1{}'\pi_{s^{(2)}[j]}$ and $\lambda\beta\pi_{s^{(2)}[j]}$ are functions in two sequence elements that can be tabulated. Equation (4) asserts that either $s^{(2)}[j]$ is a descendant of $s^{(1)}[i]$ or it is not. Recursion (3) can be verified using the recursive relationships $p'\ 1 = \lambda\beta p'_k$, $p_{k\,+\,1} = \lambda\beta p_k$, for $k \geqslant 1$, and $\pi_{s^{(n)}[i:j]} = \pi_{s^{(n)}[i:j\,-\,1]}\pi_{s^{(n)}[j]}$. $R_{i,j}$ is subject to the initial condition $R_{i,j} = 0$ if i or j is zero.

Insertion of (4) into (3) and simplification yields:

$$\begin{aligned} P(s_j^{(2)}|s_i^{(1)}) =& P'_0 P(s_j^{(2)}|s_{i-1}^{(1)}) + \lambda\beta\pi_{s^{(2)}[j]}P(s_{j-1}^{(2)}|s_i^{(2)}) \\ &+ (p_1 P_{s^{(1)}[i],s^{(2)}[j]} + p'_1\pi_{s^{(2)}[j]} \\ &- \lambda\beta\pi_{s^{(2)}[j]}p'_0)P(s_{j-1}^{(2)}|s_{i-1}^{(1)}) \end{aligned} \quad (5)$$

Again $p_1 P_{s^{(1)}[i],s[j]^{(2)}} + p'_1\pi_{s^{(2)}[j]} - \lambda\beta\pi_{s^{(2)}[j]}\ p'_0$ and $\lambda\beta\pi_{s^{(2)}[j]}$ can be tabulated, simplifying and accelerating the recursion.

Recursion (5) allows an efficient summation over all alignments of $s^{(1)}$ with $s^{(2)}$. In this context there are two additional quantities of interest, as follows below. To cope with these, it is advantageous to continue with recursions (3) and (4).

First, it is of interest to find the alignment that contributes the most to $P(s^{(2)}|s^{(1)})$, and which, given a set of parameters, will be the most probable. Secondly, it is of interest to generate alignments in

A) Parent nucleotide survives

Number of offspring from $s^{(1)}[i]$	Prefix	Tail	Substitutional evolution	Insertion
1	$P(s_{j-1}^{(2)} \mid s_{i-1}^{(1)})$	p_1	$P_{s^{(1)}[i],s^{(2)}[j]}$	1
2	$P(s_{j-2}^{(2)} \mid s_{i-1}^{(1)})$	p_2	$P_{s^{(1)}[i],s^{(2)}[j-1]}$	$\pi_{s^{(2)}[j]}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
j	$P(\emptyset \mid s_{i-1}^{(1)})$	p_j	$P_{s^{(1)}[i],s^{(2)}[1]}$	$\pi_{s^{(2)}[2:j]}$

The product of the factors in the second row corresponds to the probability of the alignment:

$$\begin{array}{cccccc} s_{i-1}^{(1)} & \star & s^{(1)}[i] & \star & - & \\ s_{j-2}^{(1)} & \star & s^{(2)}[j-1] & \star & s^{(2)}[j] & \star \end{array}$$

B) Parent nucleotide dies

Number of offspring from $s^{(1)}[i]$	Prefix	Tail	Substitutional evolution
0	$P(s_{j}^{(2)} \mid s_{i-1}^{(1)})$	p'_0	1
1	$P(s_{j-1}^{(2)} \mid s_{i-1}^{(1)})$	p'_1	$\pi_{s^{(2)}[j]}$
2	$P(s_{j-2}^{(2)} \mid s_{i-1}^{(1)})$	p'_2	$\pi_{s^{(2)}[j-1:j]}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
j	$P(\emptyset \mid s_{i-1}^{(1)})$	p'_j	$\pi_{s^{(2)}[1:j]}$

The product of the factors in the third row corresponds to the probability of the alignment:

$$\begin{array}{cccccccc} s_{i-1}^{(1)} & \star & s^{(1)}[i] & \star & - & & - & \\ s_{j-2}^{(1)} & \star & - & & s^{(2)}[j-1] & \star & s^{(2)}[j] & \star \end{array}$$

Figure 2. The independence of the indel process and the substitution process allows the two to be combined easily. (a) The possible fates of $s^{(1)}[i]$ in $s_j^{(2)}$, given that it survives. (b) The possible fates of $s^{(1)}[i]$ in $s_j^{(2)}$, given that it dies.

proportion to their probability. Methods for this are shown in an Appendix I.

Results

The method is illustrated on human α globin and β globin, that are 141 and 146 amino acids long, respectively (Figure 3). The expected length of a sequence in the equilibrium process is 143.5024, very close to the average length of the two proteins. The asymptotic variances and covariances of the parameters can be obtained from the inverse of the matrix of second derivatives of the likelihood with respect to the parameters (not shown) (Edwards, 1972).

The expected number of events in the evolution from α globin to β globin is difficult to compute, as computation of the expected number of events for any small alignment block is difficult. Take for example an amino acid aligned with another. The expected number of events would involve summing over cases where amino acids were inserted, experienced mutations and were then deleted.

However, the analogous quantity for a randomly chosen sequence in the equilibrium distribution for the estimated parameters can be calculated. Let π_i and Q_{ij} have the same meanings as in the section on substitutional models, except that the i and j terms refer to entire sequences instead of nucleotides and amino acids. For estimated time and rates, the expected number of events is $-t\ \Sigma_{i \in S}\ \pi_i Q_{ii}$. The summation is here over the complete sequence space, and π_i is the probability of i in the stationary distribution on the complete sequences, not single elements. $-Q_{ii}$ is the rate of events leaving i. The sum is equal to:

$$\frac{\mu}{\mu - \lambda}\lambda + (\mu + s)\frac{\lambda}{\mu - \lambda} = (2\mu + s)\frac{\lambda}{\mu - \lambda}.$$

This is identified as the expected number of I terms times their intensity parameters, plus the expected number of D terms times their intensity parameters, plus the expected number of S terms times their intensity parameters. It is intuitively reasonable that the expected number of insertions must equal the expected number of deletions. For the maximum likelihood parameters of this example, the expected number of insertions and deletions in the equilibrium process is 10.74. The expected number of substitution events (in the equilibrium process) is 131.59, slightly less than one event per position. The maximally contributing

Table 1. Summary of recursions

Elementary parsimony algorithm:

$$D_{i,j} = \min \begin{cases} D_{i-1,j} + g \\ D_{i-1,j-1} + d(s^{(1)}[i],\ s^{(2)}[j]) \\ D_{i,j-1} + g \end{cases}$$

Original TKF91 recursion:

$$L^0(s_i^{(1)}, s_j^{(2)}) = \frac{\lambda}{\mu}\pi_{s^{(1)}[i]}p'_0 \sum_{k=0}^{2} L^k(s_{i-1}^{(1)}, s_j^{(2)})$$

$$L^1(s_i^{(1)}, s_j^{(2)}) = \frac{\lambda}{\mu}\pi_{s^{(1)}[i]}(P_{s^{(1)}[i],s^{(2)}[j]}p_1 + \pi_{s^{(2)}[j]}p'_1) \sum_{k=0}^{2} L^k(s_{i-1}^{(1)}, s_{j-1}^{(2)})$$

$$L^2(s_i^{(1)}, s_j^{(2)}) = \pi_{s^{(2)}[j]}\lambda\beta \sum_{k=0}^{2} L^k(s_i^{(1)}, s_{j-1}^{(2)})$$

Simpler recursion:

$$P(s_j^{(2)}|s_i^{(1)}) = p'_0 P(s_j^{(2)}|s_{i-1}^{(1)}) + \lambda\beta\pi_{s^{(2)}[j]}P(s_{j-1}^{(2)}|s_i^{(1)}) + g(s^{(1)}[i], s[j])P(s_{j-1}^{(2)}|s_{i-1}^{(1)})$$

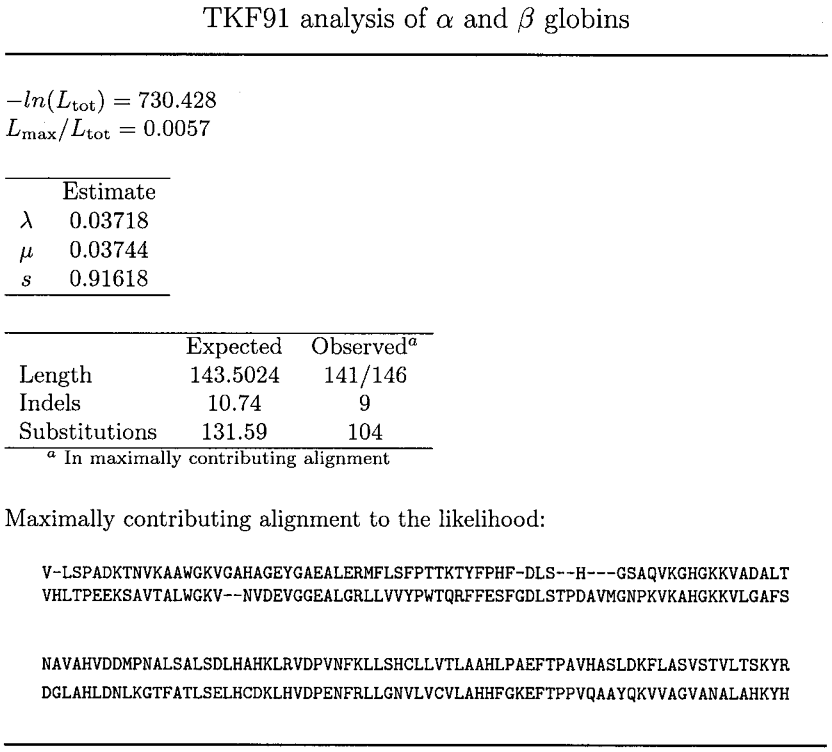
with

$$g(i,j) = p_1 P_{s^{(1)}[i],s^{(2)}[j]+(p'_1-\lambda\beta)\pi_{s^{(2)}[j]}}$$

The first recursion is the simplest parsimony algorithm. $D_{i,j}$ is the distance between $s_i^{(1)}$ and $s_j^{(2)}$, $d(\ ,\ )$ is a distance function on single elements and g is the gap penalty for a single element. All the involved quantities are integers. The second set of recursions are from the original TKF91 paper. The last recursions are from the present paper.

alignment has nine gap signs, 104 mismatches and 37 matches. Just inspecting this alignment for events would probably underestimate the number of these events in the true history of the sequences. Obviously, indels are much rarer than substitutions.

It is now possible to evaluate whether the length of sequence $s^{(2)}$, has evolved more or less than expected. The TKF91 model assumes that insertion and deletion rates are independent of sequence length. Figure 4 illustrates this distribution and β globin, for instance, is not extreme. If t goes toward



TKF91 analysis of α and β globins

$-ln(L_{\text{tot}}) = 730.428$
$L_{\text{max}}/L_{\text{tot}} = 0.0057$

	Estimate
λ	0.03718
μ	0.03744
s	0.91618

	Expected	Observed[a]
Length	143.5024	141/146
Indels	10.74	9
Substitutions	131.59	104

[a] In maximally contributing alignment

Maximally contributing alignment to the likelihood:

```
V-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLS--H---GSAQVKGHGKKVADALT
VHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLGAFS


NAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR
DGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH
```

Figure 3. α Globin and β globin analysis using the TKF91 model.

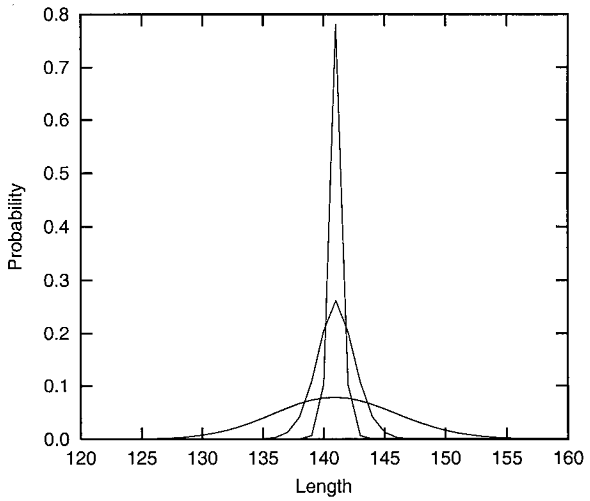


Figure 4. The length distribution of a protein that evolves from α globin with the estimated parameters of globin evolution (for 20, 200, and 2000 million years). For derivations see Appendeses. For instance, the length of the β globin (146) is not very extreme in the distribution of distance lengths, if starting with 141 amino acids and evolving for 800 million years. This is twice a resonable guess of the time to the most recent common ancestor of α and β globin.

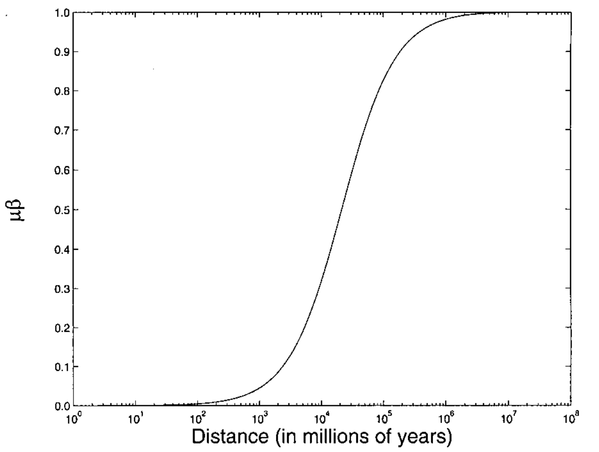


Figure 5. The β function plotted with parametersestimated from the α, β globin analysis. At the time when μ β is 0.5, the descendants of the immortal link are expected to contribute half of the complete sequence. This time is seen to bearound 20 billion years. The time taken for it to contribute 5 % is around one billion years. The effect of the immortal link on realsequences is vanishing over realistic time periods.

infinity, this distribution will go towards the equilibrium length distribution for this process, but very slowly. Specifically, the mortals and their descendants will go extinct, while the descendants of the immortal link will be dominating the complete sequence. This means that $p_k''(t)$ is a geometric distribution with parameter $\beta\lambda$, when $t \to \infty$. As shown in Figure 4, even if the most recent common ancestor were 2000 million years back in time, the length is still distributed as a bell around the initial length. If a sequence is chosen from the equilibrium distribution and observed for a period of time, the descendants of the immortal links will be expected to constitute $\mu\beta(t)$ of the complete sequence. $\mu\beta(t)$ is plotted in Figure 5 and it is clear that it converges very, very slowly to one.

Computational results

Computationally, maximum likelihood alignment is inherently more expensive than parsimony/similarity. There are three areas of importance to the time of performing a likelihood alignment: the number of entries in the matrix needing to be calculated, the number of evaluations needed to find the maximum likelihood estimate and the time used in calculating the basic recursion

Matrix entries necessary

Figure 6 illustrates the alignment path of α globin and β globin including boundaries defined by suboptimal alignments. (In these investigations PAM250 was used for similarity alignments and a gap penalty cost of 4.5 was used per amino acid.) For a given ε, a boundary corresponding to the suboptimal paths with a score of $1 - \varepsilon$ of the optimal score can be found. This defines an area of the matrix. If ε is less than zero, the area is empty, if it is zero, it will be the entries that are on optimal alignments of the sequences. As ε increases, the defined area will converge toward the complete matrix. Figure 7 shows how much of the likelihood function is found within the area as a function of ε. The sequences involved were derived from a sequence of length 1500 amino acids, that experienced evolution corresponding to the difference between α globin and myoglobin. It can be observed that the relative underestimation of the likelihood is less than $e^{-13} = 2.3 \times 10^{-6}$ if an ε of 0.01 is used. An ε value of 0.01 corresponds to 1.8×10^{-3} of the area of the complete matrix. This gives rise to a very significant acceleration. This is a favourable case, but in general the acceleration is considerable and the area containing all significantly contributing paths is very narrow. The closer related the sequences are, the narrower the band will typically be.

If ε is too small, alignments that contribute significantly to the alignment will be discarded, resulting in a bias in the estimated parameter values. Since a low ε value will discard alignments with many indels, this is expected to create a bias towards smaller values of μ and λ, which was also observed (results not shown).

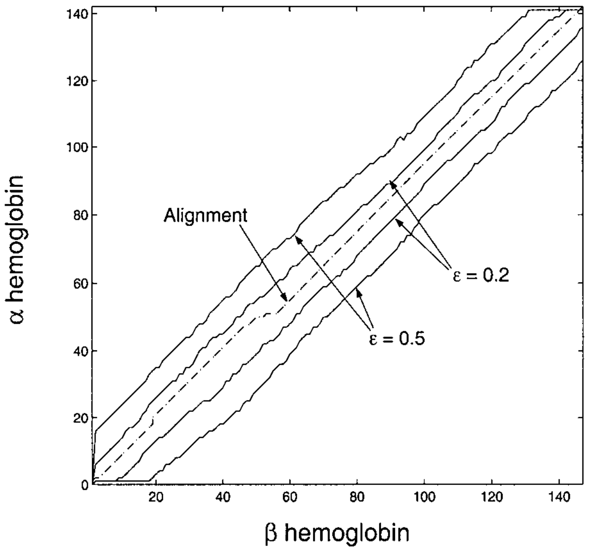


Figure 6. Illustration of the similarity alignment as a path in the matrix. The maximally contributing statistical alignment and the similarity alignment is identical in this simple case. The area containing nodes that could be on a suboptimal solution, better than (1 − ε) of the optimal similarity score, is also illustrated for ε equal to 0.2 and 0.5. For practical purposes a band much narrower can be used, typically less than 0.005.

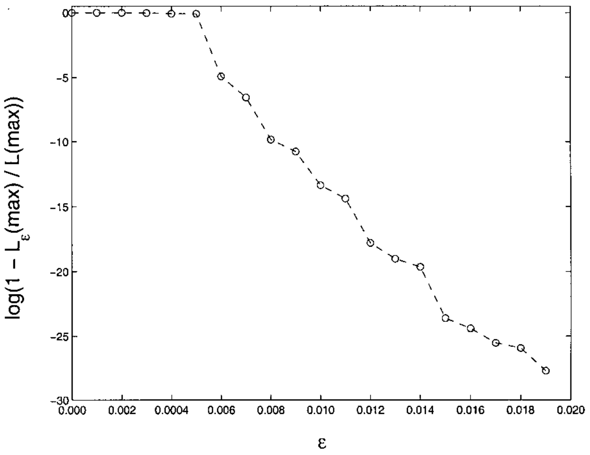


Figure 7. A plot of $\log(1 - L_\varepsilon/L)$ as a function of ε for simulated globins of length 1500, with evolutionary distance like myoglobin (153 amino acids) and α globin (141 amino acids). This maps [0, 1] into [0, − ∞], and illustrates how much of the contributions to the likelihood function is within ε of the similarity optimum alignment solution. It is obvious that most contributing paths to the likelihood functions are within a very narrow band. This points to an obvious speedup of the likelihood method.

The number of evaluations

This will consist of three parts: an initial guess of parameters, an algorithm searching for the minimum, and a stopping criteria determining whether the current parameter estimates are sufficiently close to the maximum likelihood values. The problem is illustrated in Figure 8, where the $L(\mu,s)/L_{\max}$ is plotted in an area close to the maximum likelihood estimate for α globin and myoglobin. The method used is to make a good guess that is within the through containing the maximum likelihood point, crawl in few steps to the bottom of the basin and stop, when iterations does not improve the estimates significantly.

Initial guess. We only consider the protein model and the strategy will be to assume that the parsimony/similarity alignment is the correct alignment. We calculate how many gap signs, matches, and mismatches would be expected and choose the parameters that give these expectations. In the protein substitution model, the expected number of mismatches, $(1 - \Sigma_i \pi_i P_{ii}) n_{\text{align}}$, were calculated and equated to the observed number, giving a guess for s (n_{align} is the number of columns without gap signs in the alignment). λ and μ was guessed by first observing that the expected length of a sequence is $\lambda/(\mu - \lambda)$. This would fix the ratio of λ/μ to:

$$\frac{\lambda}{\mu} = \frac{l_{\text{ave}}}{l_{\text{ave}} + 1}$$

where l_{ave} is the average sequence length. In addition, it is possible, when knowing the length of one sequence and the p functions, to calculate the expected number of gap signs in an alignment to (see Appendices):

$$\#\text{gap} = \frac{\lambda\beta}{1 - \lambda\beta} + s\Bigg(e^{-\mu t}\frac{\lambda\beta}{(1 - \lambda\beta)} + \mu\beta + (1 - e^{-\mu t} - \mu\beta)^2\frac{2 - \lambda\beta}{1 - \lambda\beta}\Bigg)$$

Using this, a guess of $\mu = 0.0316$ and $s = 0.9500$ is obtained from the α globin *versus* β globin similarity alignment. A slightly inferior guess of μ can be obtained by assuming that the observed gap signs are the events that actually have happened in the evolutionary history. This gives $2\mu L = \#\text{gap}$ and will give a lower estimate than using the p functions. Using this method μ is estimated to 0.0311. This difference is probably larger for more distantly related sequences. The maximum likelihood results are shown in Figure 3.

Optimisation. Several numerical optimisation methods were tried (e.g. simplex and Powell), but given a good initial guess, the best was BFGS (Broyden-Fletcher-Goldfarb-Shanno) (Press *et al.*, 1992). An example of the search for the maximum likelihood estimate is illustrated in Figure 8, for simulated sequences approximately of length 1500,

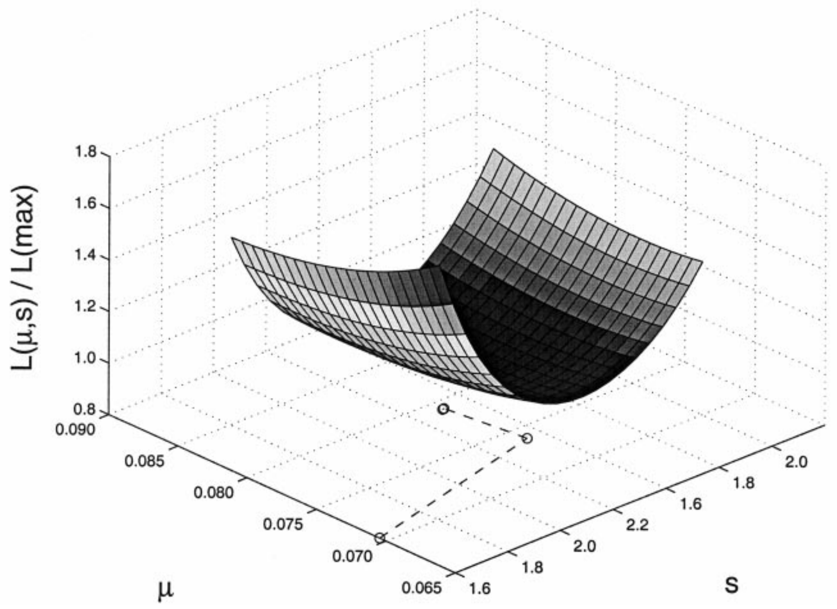


Figure 8. Likelihood surface for human α globin aligned with human myoglobin. Only two parameters are allowed to vary, since the ratio of λ/μ has been fixed to $l_{\text{ave}}/(l_{\text{ave}}+1)$. l_{ave} is the average length of the two proteins. The numerical optimisation part of the statistical alignment problems is to find an initial point within this valley, as close as possible to the bottom, and then through a series of iterations get close to the minimum. In the floor of the diagram the search for the minimum is shown. $(s, \mu)_0$ is the initial guess obtained from analysing the similarity alignment. After three iterations the improvements in the likelihood was negligible. BFGS (see the text) had then used 28 evaluations of the likelihood function. Each iteration needs several evaluations to determine first and second derivatives of the likelihood function (in our implementation derivatives were found numerically, but could in principle be found by dynamical programming).

with an evolutionary distance like α globin and myoglobin. In this case, four iterations and 28 likelihood evaluations were needed. Each iteration needs a series of likelihood evaluations to determine first and second derivatives of the likelihood function. The total number of likelihood evaluations is typically less than 50.

Stopping condition. When an iteration produced changes in the likelihood estimates, that was less than 10^{-3}, the iterations stopped. Figure 9 shows $L_{\text{tot}}(k)$ as function of iteration number, k. It can be seen that major improvements are obtained in the first few jumps. After three to four iterations $L_{\text{tot}}(k)$ is very close to the likelihood function taken in the maximum likelihood estimate.

The basic recursion

The likelihood recursions (three to four) of TKF91 are a bit more complicated than the recursion in the optimisation alignment algorithm (parsimony/similarity). Comparison between the two indicated that the likelihood recursion was 50-70 times slower than the optimisation recursion. The main reason for this large difference is that multiplication of reals is slower than addition of integers on most computers.

Summary

The above improvements yield a method that is significantly faster than the one described by Thorne *et al.* (1991). It seems probable that a further increase in speed can be obtained from focusing on the last two factors. In absolute terms, two proteins of length 1500, can be analysed in less than five seconds on a fast desktop computer (Silicon Graphics Octane with a 300 MHz R12000 processor), which makes statistical alignment a fully practical method for two sequences.

Homology test

Consider the α globin and myoglobin. Are they homologous? Homology must here be the answer to the question, whether the value of t is finite or infinite. A value of infinity implies that they could both have been drawn independently from the stationary distribution of the evolutionary process. Statistical alignment can contribute considerably to this question.

Parsimony/similarity alignment based test

Most tests in a parsimony/similarity alignment setting presuppose an alignment and regard the matched positions as independent realizations of the same distribution.

Most homology tests are based on a similarity scoring function, for each position in the alignment, of the form: $W_{i,j} = \ln(\pi_i P^{2.5}_{i,j}/(\pi_i \pi_j))$ (Altschul, 1993). In this expression, $P^{2.5}_{i,j}$ is the transition probabilities, when 2.5 units of time has passed. This amounts to choosing among the competing hypothesis that two sequences are 2.5 events apart *versus* infinitely far apart. It only handles substitutions "correctly". The rationale for indel cost is more arbitrary.

In a frequently used test, the shuffle test, the two sequences are aligned and a score is obtained (Doolittle, 1986). The significance of this score is

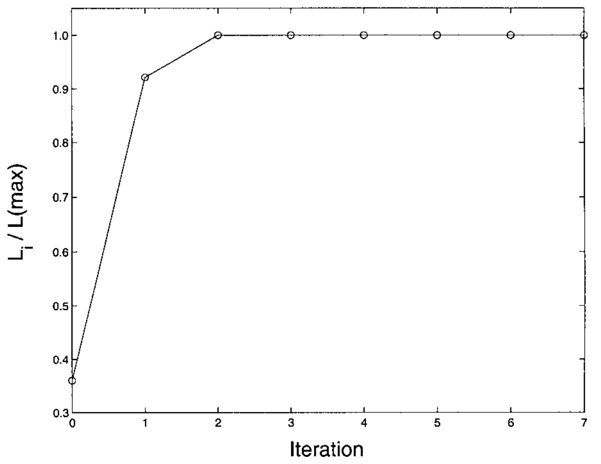


Figure 9. This Figure shows the likelihood values in different iterations. After three to four steps, a guess very close to the maximum likelihood has been achieved.

evaluated by permuting the order of the amino acids and aligning the permuted proteins:

$$\begin{array}{cc} \text{Real} & \text{Random} \\ s^{(1)} = ATWYFCAK\text{-}AC & s^{(1)} = ATWYFC\text{-}AKAC \\ s^{(2)} = ETWYKCALLAD & s^{(2)} = LTAYKADCWLE \\ {*}{*}{*}\ \ {*}{*}\ \ \ {*} & {*}{*} \end{array}$$

This is done many times and the real score is compared to the score of the permuted proteins. This amounts to sampling in the observed amino acid distributions without replacement. If the score for the real sequences are much better (high for similarity alignment or low for parsimony alignment), the proteins are assumed homologous. An illustration of this test for α globin and myoglobin is shown in Figure 10 (top).

This approach has several drawbacks. Firstly, it is dependent on having the correct alignment, which is unlikely for distant sequences. Secondly, it must fix a time back to a common ancestor that is unknown, since any substitution matrix assumes a distance between sequences. Thirdly, it is hard to introduce more realistic models of sequence evolution in this test. Statistical alignment has the potential of solving these problems.

Statistical alignment based test

In testing homology we are asking if two sequences have a common ancestor finitely far back in time. Here we try to distinguish two competing hypotheses. Are they independent sequences from the equilibrium distribution on the set of sequences? Or are they related by a tree with a root finitely far back in time? The test will be parametric bootstrap as described by Cox (1962).

(1) All parameters, $(\lambda_{\text{real}}, \mu_{\text{real}}, s_{\text{real}})$, are estimated by maximum likelihood for the given sequences.

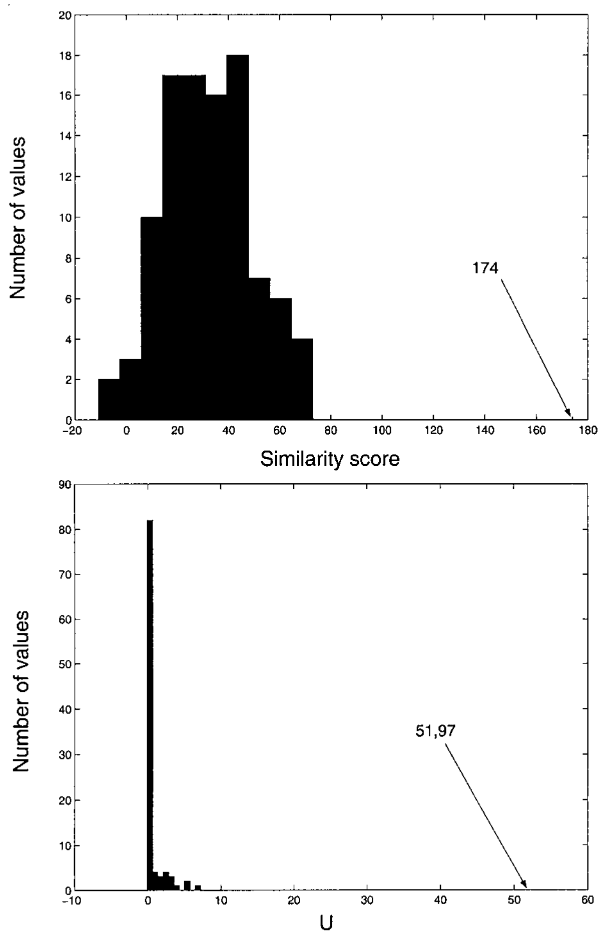


Figure 10. Top: shuffle testing of the homology between myoglobin and α globin. The arrow to the right is the score of the real sequences. Bottom: testing the homology between myoglobin and α globin, using statistical alignment. The arrow to the right is the score of the real sequences.

(2) Pairs of independent sequences, $(s^{(1)}, s^{(2)})_i$, are simulated using these parameters.

(3) These simulated sequences are analysed using statistical alignment and parameters are reestimated, (λ_i, μ_i, s_i).

The following statistic, U, is calculated for the real sequences and for the simulated sequences:

$$U = -2\ln\frac{P(s^{(1)}, s^{(2)})}{P(s^{(1)})P(s^{(2)})}$$

Now, the value for the real sequences can be compared to the distribution of the values for the simulated sequences. If the value for the real sequences is extreme for the distribution, the sequences are homologous.

An illustration of this test for α globin and myoglobin is shown in Figure 10 (bottom). This approach has solid potential, but at present it has a number of drawbacks that prevent it from broad use. Firstly, it is much slower than database scanning programs. Secondly, the TKF91 process is not a realistic model for sequence evolution. Especially, the geometric equilibrium distribution of sequence lengths is not believable and should be improved.

A method for homology testing, that also sums over all alignments, but without an evolutionary model has been made by Bucher & Hofmann (1996).

Goodness of fit-testing the TKF process

It is obviously of importance to test the proposed model when using it to analyse real data. Tests have been developed (especially due to Goldman (1993)) for testing substitution models, when an alignment is given. Since the new aspect of the TKF91 process is the indel process, we will focus on testing whether this aspect of sequence evolution is well modelled. The TKF91 model assumes that insertions and deletions occur in steps of one.

Many optimisation alignment methods put much emphasis on having the correct gap penalty function and assume that indels can involve longer segments. The alignments obtained by the TKF91 method can also have consecutive runs of gaps signs, but they would all have been inserted or deleted individually. If longer indels occur, it should nonetheless be reflected in longer runs of gap signs in the alignments proposed by the TKF91 method. It is therefore very natural to compare the p functions with the corresponding configurations of survival and number of descendant obtained from the TKF91 method.

Again, consider human α globin and β globin. If their true alignment could be observed, the fate of 141 amino acids and one immortal link had been observed. Table 2B shows which numbers would be obtained if the alignment in Figure 3 were used. Given the maximum likelihood parameters and the two sequences, Table 2C can be filled by a dynamic programming algorithm, using recursion (1-2) to assign probabilities of the fate of $s^{(1)}[i]$ in $s^{(2)}$ given that $\{s_i^{(1)} \to s_j^{(2)}\}$. We chose to sample alignments (100) according to their probability, using the stochastic backtracking procedure, since this was easier to program. This is not an alignment chosen uniformly among all possible alignments, but chosen randomly in proportion to how much they contribute to the likelihood function in the maximum likelihood point.

The difference between Table 2A and C is measured by the $X^2 = (\text{obs} - \text{exp})^2/\text{exp}$ statistic. This is 532.17 in this case. The cells contributing to this are shown in Table 2D. It is obvious from alignments of real sequences, that longer runs of gap signs occur, that are not in accordance with the model.

It is possible to get longer series of gap signs in alignments of real sequences, that contribute massively to the X^2_{real} statistic. If this contribution is statistically significant, a natural interpretation would be that longer indels had occurred. Nonetheless, alternative explanation cannot be ruled out. For instance, the indel rate could be unevenly distributed along the sequence, so many single indels occurring next to each other were actually quite probable.

To assess significance between Table 2A and C, 100 sequences (s_i terms) were simulated starting from α globin and evolving according to the estimated evolutionary parameters. The X^2 statistics were calculated from these by making analogues

Table 2. Results from the goodness of fit test for the indel model in the evolution of α globin to β globin

A. *Expected according to model and length of* α *globin*							
No. of descendants	0	1	2	3	4	5	6
Immortal link	-	0.9642	0.0346	0.0012	0.0000	0.0000	0.0000
Mortal links - survived	-	130.95	4.6937	0.1682	0.0060	0.0002	0.0000
Mortal links - died	5.0891	0.0890	0.0032	0.0004	0.0000	0.0000	0.0000
B. *Observed in optimal alignment*							
No. of descendants	0	1	2	3	4	5	6
Immortal link	-	1	0	0	0	0	0
Mortal links - survived	-	135	2	1	1	0	0
Mortal links - died	2	0	0	0	0	0	0
C. *Expected according to model and sequences*							
No. of descendants	0	1	2	3	4	5	6
Immortal Link	-	0.95	0.05	0.00	0.00	0.00	0.00
Mortal links - survived	-	132.77	4.14	0.67	0.35	0.21	0.05
Mortal links - died	2.68	0.12	0.01	0.00	0.00	0.00	0.00
D. X^2 *difference between A and C*							
No. of descendants	0	1	2	3	4	5	6
Immortal Link	-	0.0004	0.0069	0.0012	0.0000	0.0000	0.0000
Mortal links - survived	-	0.0253	0.0653	1.496	19.62	203.62	311.04
Mortal links - died	1.1404	0.0108	0.0145	0.0001	0.0000	0.0000	0.0000

Maximum likelihood parameter estimates were obtained from analysis of α globin and β globin and then regarded as fixed. Each section (A to D) tabulates the quantities relating to the three p functions. A. The expectation from the different p functions. For the mortal links, these expectations are $141p_k$ (survived) and $141p'_k$ (died). For the immortal link it is simply p_k'', since there is only one. B. The result, if the optimal alignment (Figure 3) were used in filling out the Table. C. Sampling 100 random alignments in proportion to their probability using equation (11). D. Contributions to the X^2 statistic form the difference between A and C.

to Table 2A and C. If X^2_{real} is extreme in this distribution, the indel process does not fit well with the real sequences. The distribution of X^2 is shown in Figure 11 together with X^2_{real}. Obviously, the TKF91 needs modification to be a satisfactory model.

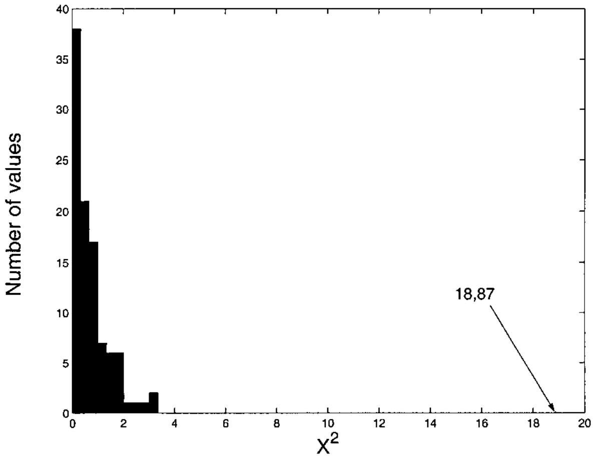


Figure 11. Goodness of fit testing indel lengths in the TKF model. The X^2 value of the real data (marked by an arrow) is very extreme in the distribution of the simulated X^2 values.

Discussion

This article has highlighted some of the advantages of a more statistical approach to alignment, the main ones being:

(1) It is explicitly founded on a description of molecular evolution.

(2) Parameters are estimated and biologically meaningful.

(3) Different evolutionary events can be assigned probabilities.

However, the present model is unrealistic, thus, many generalisations and improvements are of immediate practical interest.

Since it is clear that indels longer than one nucleotide or amino acid do occur, incorporating this into a model would be a significant step towards biological realism. However, it is not straightforward to do this. Allowing for longer insertions is simple and such a longer insertion could be associated to a single link, as in the TKF91 process. Longer deletions remove intervals of sequences, and should be modelled so that the whole process is time reversible, since this has computational advantages. There is no biological reason for believing that the insertion process should be the time-reversed image of the deletion process. It remains to be explored how seriously this assumption is violated in real data.

A second extension would be to generalise the TKF91 dynamic programming algorithm, calculating the likelihood for a set of homologous sequences. Steel & Hein (2000) have done this for k sequences, related by a star-shaped tree. J.H. (unpublished results), has generalised this further to k sequences, related by a binary tree, in an algorithm that has $O(l^k)$ running time in the sequence length. This is analogous to the parsimony algorithm relating k sequences devised by Sankoff (1975). However, an implementation of the likelihood method would be much slower than the parsimony/similarity method, due to its more complicated algorithm, parameter optimisation, etc. To yield a practical statistical multiple alignment method, other methods than the dynamic programming algorithm would have to be used, e.g. Markov chain Monte Carlo methods.

Modelling substitutions and indels that are unevenly distributed along the sequences could also be improved. Real sequences will have different probabilities of insertion/deletion for different regions. For proteins, it is well known that insertion/deletions are more frequent in loop regions than in sheets and helices, and it would give a more realistic model to take advantage of this knowledge. Incorporating the hidden Markov model for different structural categories as done by Goldman *et al.* (1996) seems especially relevant. Using hidden Markov models will pose a problem, since the indel process would make the Markov model longer and shorter at stochastic times.

The TKF91 model is simple and tractable, but it would be of interest to explore alternatives. The view of a sequence being tagged by an immortal link does not conform to biological intuition. A possibility would be to let a sequence be born from a given equilibrium distribution and to be killed according to some process. Whether this could lead to a tractable process remains to be explored, but it would conform better to biological intuition and could give a better equilibrium distribution of sequence lengths than the geometric distribution of the TKF91 model. In this context, it would also be of interest to formulate how subsequences can be homologous to subsequences. The tests described here were solely addressed in terms of global comparisons and to devise a practical competitive test, it would be necessary to formulate an analogue of local alignment for statistical alignment.

More realistic models of sequence evolution and methods for aligning more sequences would automatically lead to better homology tests. In this context, it should be noted that when molecular biologists perform homology tests (or database searches), their prime objective is not homology, but rather inferences about function. It might be advantageous to model this explicitly, i.e. to model not only the sequence but also the probability that a sequence with one function obtains another function. The approach taken here might also unify the contending approaches of Dayhoff *et al.* (1978) *versus* Henikoff & Henikoff (1992), in making score matrices. Dayhoff constructs matrices from closely related sequences that will define log-odd scores for distantly related sequences. Henikoff & Henikoff use conserved blocks in distantly related sequences to define log-odd scores directly. These two approaches seem to focus on quickly and slowly evolving positions, respectively. A statistical alignment model directly incorporating quickly and slowly evolving positions would unify the two approaches.

The concept of homology in sequence comparison is not crystal clear. Since the earliest organism probably contained very few sequences (possibly only one), maybe all sequences are homologous in the strict sense. There have been assertions about the number of different protein families appearing in life on earth (Chothia, 1992).

Statistical approaches to alignment have many advantages relative to parsimony/similarity approaches, but the latter methods have a large lead in software developments. Even if statistical approaches were developed to a stage where it was better at the conceptual and good at the algorithmic level, there would still be a huge software gap for many years to come.

Comment

The programs and tests developed in this paper can all be accessed at the web-site: www.brics.dk/ ~ compbio. The program contains the following parameters to be set of the user: the narrowness of the band where dynamical programming is performed and the level of precision in parameters, when iterations are to be stopped.

Acknowledgements

Anne-Mette Krabbe Pedersen, Christian Nørgård Storm Pedersen, Jan Gorodkin, Mikkel Schierup, Jakob Skou Pedersen, Joseph Felsenstein, and the anonymous reviewer are thanked for useful comments on the manuscript. C.W. and J.H. were generously supported by the Newton Institute of Mathematical Sciences. G.W. was supported by BRICS and Danish Agricultural and Veterinary Research Council (grant 9901522). C.W. was supported by grant BBSRC 43/MMI09788 and by the Carlsberg Foundation.

References

Allison, L. & Wallace, C. S. (1994). The posterior probability distribution of alignments and its application to parameter estimation of evolutionary trees and to optimization of multiple alignments. *J. Mol. Evol.* **39 (4)**, 418-430.

Altschul, S. F. (1993). A protein alignment scoring system sensitive at all evolutionary distances. *J. Mol. Evol.* **36 (3)**, 290-300.

Bishop, M. J. & Thompson, E. A. (1986). Maximum likelihood alignment of DNA sequences. *J. Mol. Biol.* **190 (2)**, 159-165.

Bucher, P. & Hofmann, K. (1996). A sequence similarity search algorithm based on a probabilistic interpretation of an alignment scoring system. In *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology* (States, D. J., Agarwal, P., Gaasterland, T., Hunter, L. & Smith, R. F., eds), pp. 44-51, AAAI Press, California.

Chothia, C. (1992). One thousand families for the molecular biologist. *Nature,* **357 (6379)**, 543-544.

Cox, D. (1962). Further results on tests of separate families of hypotheses. *J. R. Stat. Soc. ser. B,* **24**, 406-424.

Dayhoff, M. O., Schwartz, R. M. & Orcutt, B. C. (1978). A model of evolutionary change in proteins, matrices for detecting distant relationships. In *Atlas of Protein Sequence and Structure* (Dayhoff, M. O., ed.), vol. 5, pp. 345-352, Cambridge University Press, Washington, DC.

Doolittle, R. F. (1986). *Of URFs and ORFs: A Primer on How to Analyze Derived Amino Acid Sequences*, University Science Books, California.

Edwards, A. W. F. (1972). *Likelihood*, Cambridge University Press, Cambridge.

Goldman, N. (1993). Statistical tests of models of DNA substitution. *J. Mol. Evol.* **36 (2)**, 182-198.

Goldman, N., Thorne, J. L. & Jones, D. T. (1996). Using evolutionary trees in protein secondary structure prediction and other comparative sequence analyses. *J. Mol. Biol.* **263**, **(2)**, 196-208.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol.* **162**, 705-708.

Henikoff, S. & Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA,* **89 (22)**, 10915-10919.

Mitchison, G. J. (1999). A probabilistic treatment of phylogeny and sequence alignment. *J. Mol. Evol.* **49 (1)**, 11-22.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (1992). *Numerical Recipies in C*, 2nd edit., Cambridge University Press, Cambridge.

Sankoff, D. (1975). Minimal mutation trees of sequences. *SIAM J. Appl. Math.* **28**, 35-42.

Steel, M. & Hein, J. J. (2000). A generalisation of the Thorne-Kishino-Felsenstein model of statistical alignment to *k* sequences related by a star tree. *Appl. Math. Letters,* **in the press**.

Thorne, J. L., Kishino, H. & Felsenstein, J. (1991). An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol.* **33 (2)**, 114-124.

Thorne, J. L., Kishino, H. & Felsenstein, J. (1992). Inching toward reality: an improved likelihood model of sequence evolution. *J. Mol. Evol.* **34 (1)**, 3-16.

Zhu, J., Liu, J. S. & Lawrence, C. E. (1998). Bayesian adaptive sequence alignment algorithms. *Bioinformatics,* **14 (1)**, 25-39.

Appendix I: Finding Alignments

The most probable alignment for a given set of parameters, can be found. Since the number of alignments is large, this probability will be small relative to $P(s^{(2)}|s^{(1)})$. The reasoning behind equations (3)-(4) can be applied again. Let $s^{(1)} \to s^{(2)}$ denote all evolutionary paths from $s^{(1)}$ to $s^{(2)}$. Instead of keeping track of the set $\{x \in s_i^{(1)} \to s_j^{(2)}\}$ and $\{x \in s_i^{(1)} \to s_j^{(2)}|s^{(2)}[j]$ is a descendant of $s^{(1)}[i]$ in $x\}$ with many alignments in them, only keep the most probable alignment in these sets (indicated with a $\{\ \}_{\max}$):

$$R^{\max}_{i,j} = \max\{p_1 P_{s^{(1)}[i],s^{(2)}[j]} P(\{s^{(1)}_{i-1} \rightarrow s^{(2)}_{j-1}\}_{\max}), \quad p'_1 \pi_{s^{(2)}[j]} P(\{s^{(1)}_{i-1} \rightarrow s^{(2)}_{j-1}\}_{\max}),\ \lambda\beta\pi_{s^{(2)}[j]} R^{\max}_{i,j-1}\} \tag{6}$$

$$P(\{s^{(1)}_{i} \rightarrow s^{(2)}_{j}\}_{\max}) = \max\{R^{\max}_{i,j}, p'_0 P(\{s^{(1)}_{i-1} \rightarrow s^{(2)}_{j}\}_{\max})\} \tag{7}$$

Using backtracking, the most probable alignment can be found. This alignment is of little interest and is mainly calculated to be able to generate one alignment for illustration. As shown by Thorne *et al*. (1991), this alignment is not representative of the actual history of $s^{(1)}$ and $s^{(2)}$, but without it, this method would be an alignment method that did not produce any alignment.

Alignments can be generated in proportion to their probability. This can be done by the following procedure starting in ($l1$, $l2$) (the lengths of the two sequences) and going down to (0, 0):

Step	Probability	Alignment block
$R_{i,j} \rightarrow P(s^{(2)}_{j-1}\|s^{(1)}_{i-1})$	$p_1 P_{s^{(1)}[i],s^{(2)}[j]} P(s^{(2)}_{j-1}\|s^{(1)}_{i})/R_{i,j}$	$s^{(1)}[i]$ $s^{(2)}[j]$
$R_{i,j} \rightarrow P(s^{(2)}_{j-1}\|s^{(1)}_{i-1})$	$p'_1 \pi_{s^{(2)}[j]} P(s^{(2)}_{j-1}\|s^{(1)}_{i})/R_{i,j}$	$s^{(1)}[i]$ – – $s^{(2)}[j]$
$R_{i,j} \rightarrow R_{i,j-1}$	$\lambda\beta\pi_{s^{(2)}[j]} P(s^{(2)}_{j-1}\|s^{(1)}_{i})/R_{i,j}$	– $s^{(2)}[j]$
$P(s^{(2)}_{j}\|s^{(1)}_{i}) \rightarrow R_{i,j}$	$R_{i,j}/P(s^{(2)}_{j}\|s^{(1)}_{i})$	Nothing
$P(s^{(2)}_{j}\|s^{(1)}_{i}) \rightarrow P(s^{(2)}_{j}\|s^{(1)}_{i-1})$	$p'_0/P(s^{(2)}_{j}\|s^{(1)}_{i-1})/P(s^{(2)}_{j}\|s^{(1)}_{i})$	$s^{(1)}[i]$ –

It is also possible to sample random alignments, using an analogue to equation (5), but it seems difficult to formulate a maximum analogue to equation (5) in the style of equations (6) and (7).

Reference

Thorne, J. L., Kishino, H. & Felsenstein, J. (1991). An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Enol.* **33(2)**, 114-124.

Appendix II: Calculating Length Distributions

Take a sequence of length n. Letting this sequence evolve over time, makes the length change according to the λ and μ parameters of the model. It is possible to calculate the distribution of lengths for any given time t that passes.

For low values of t, the length distribution will be very narrow around n. For larger values of t, the distribution becomes a skewed bell shape around n. With very large t values the distribution will become geometric as dictated by the λ and μ parameters.

The generating functions (GFs) can be found for the number of children of mortal and immortal links. Multiplying an appropriate number of these can give the GF for the entire length of a sequence. Thus, given an initial length and an amount of time, the length distribution can be calculated as (P_m being the probability of having length m at time t, starting at length n):

$$P_m = \sum_{i=0}^{\min(m.n)} \left(\begin{matrix} n+m-i \\ m-i \quad n-i \quad i \end{matrix}\right)(-a)^{n-i} d^{n-i} c^{i} b^{-n-m+i-1}$$

with:

$$a = a(t) = -\frac{\lambda}{\mu-\lambda}\gamma(t)$$

$$b = b(t) = 1 - a(t) = 1 + \frac{\lambda}{\mu-\lambda}\gamma(t)$$

$$c = c(t) = 1 - \frac{\lambda}{\mu-\lambda}\gamma(t)$$

$$d = d(t) = 1 - c(t) = \frac{\lambda}{\mu-\lambda}\gamma(t)$$

$$\gamma(t) = 1 - e^{(\lambda-\mu)t}$$

The average and variance of the length distribution is:

$$E(S_t) = n + \left(\frac{\lambda}{\mu - \lambda} - n\right)\gamma(t)$$

$$Var(S_t) = nc(1 - c) - (n + 1)a(1 - a)$$

Appendix III: Expected Number of Gaps

The expected number of gaps produced by a mortal link in time t, times its probability of survival is:

$$g(t) = \sum_{n=1}^{\infty} p_n(t)(n - 1)$$

since $n - 1$ gaps are produced when a mortal link has n children and it survives. This can be written as:

$$g(t) = e^{-\mu t}(1 - \lambda\beta)\sum_{n=0}^{\infty} n(\lambda\beta)^n = e^{-\mu t}\frac{\lambda\beta}{(1 - \lambda\beta)}.$$

The same calculations can be done for mortal links that die and for immortal links. The total expected number of gaps is the sum of an appropriate number of each of these g functions:

$$\#gap = \frac{\lambda\beta}{1 - \lambda\beta} + s\left(e^{0\mu t}\frac{\lambda\beta}{(1 - \lambda\beta)} + \mu\beta + (1 - e^{-\mu t} - \mu\beta)\frac{2 - \lambda\beta}{1 - \lambda\beta}\right)$$

$$\beta = \frac{1 - e^{(\lambda - \mu)t}}{\mu - \lambda e^{(\lambda - \mu)t}}$$

$$\frac{\mu t}{\lambda t} = \frac{s + 1}{s}$$

Edited by J. Karn

(Received 17 April 2000; received in revised form 21 July 2000; accepted 25 July 2000)