Rapid Significance Estimation in Local Sequence Alignment with Gaps

Ralf Bundschuh
The Ohio State University
Department of Physics
174 West 18th Avenue
Columbus, OH 43210
U.S.A.
Phone: +1 (614) 292-5713
Fax: +1 (614) 292-7557
rbund@mps.ohio-state.edu

June 5, 2001

Abstract

In order to assess the significance of sequence alignments it is crucial to know the distribution of alignment scores of pairs of random sequences. For gapped local alignment it is empirically known that the shape of this distribution is of the Gumbel form. However, the determination of the parameters of this distribution is a computationally very expensive task. We present a new algorithmic approach which allows to estimate the more important of the Gumbel parameters at least five times faster than the traditional methods. Actual runtimes of our algorithm between less than a second and a few minutes on a workstation bring significance estimation into the realm of interactive applications.
Keywords: Sequence alignment, importance sampling, statistical significance, Gumbel distribution

1 Introduction

Sequence alignment is one of the oldest and most commonly used computational tools of molecular biology. Its applications range from the identification of the function of newly sequenced genes to the construction of phylogenic trees (Doolittle 1996; Waterman 1995). Alignment algorithms come in different levels of complexity. The simplest alignment algorithm is gapless alignment as it was implemented for performance reasons in the original version of BLAST (Altschul *et al.* 1990). It is very fast and theoretically very well understood. However, in order to detect weakly homologous sequences, gaps have to be allowed in an alignment (Pearson 1991). This has been formalized already long before the appearance of BLAST in the more sophisticated Smith-Waterman algorithm (Smith and Waterman 1981). The latter has, e.g., been implemented in FASTA (Pearson and Lipman 1988) and later been taken over into the current versions of BLAST (Altschul *et al.* 1997). In order to find even more remote homologies, iterative schemes like PSI-BLAST (Altschul *et al.* 1997) have to be employed.

All alignment algorithms have the drawback that they will find an optimal alignment and an optimal score for *any* pair of sequences — even randomly chosen and thus completely unrelated ones. Thus, it is necessary to assess the significance of a resulting alignment. A popular approach to this problem is to compare the score of the optimal alignment to the scores generated by the optimal alignments of *randomly chosen* sequences. This is quantified by the p- or E-value. This comparison steadily becomes more important since with the increasing size of the databases the probability for obtaining a relatively large score just by chance increases dramatically.

In order to be able to quote a p-value the *distribution* of optimal alignment scores for alignments of random sequences has to be known. In the case of gapless alignment it has been worked out rigorously (Karlin and Altschul 1990; Karlin and Dembo 1992; Karlin and Altschul 1993) that this distribution is a Gumbel or extreme value distribution (Gumbel 1958). It is characterized by two pa-

rameters which depend on the scoring system used and on the amino acid frequencies with which the random sequences are generated. For gapless alignment also this dependence of the two Gumbel parameters on the scoring system is completely known. This precise characterization of the statistical significance made the original version of BLAST (Altschul *et al.* 1990) a very successful tool.

For the case of gapped alignment, there is no theory which describes the distribution of alignment scores of random sequences. However, there is a lot of numerical evidence that the distribution is still of the Gumbel form (Smith *et al.* 1985; Collins *et al.* 1988; Mott 1992; Waterman and Vingron 1994a; Waterman and Vingron 1994b; Altschul and Gish 1996; Olsen *et al.* 1999). Nevertheless, it has turned out to be a very challenging problem to find the two Gumbel parameters for a given scoring system. For a limited range of scoring systems, analytical approximations are available (Mott and Tribe 1999; Bundschuh 2000; Mott 2000; Siegmund and Yakir 2000). For practically relevant scoring systems however, the Gumbel parameters have to be estimated numerically. The straightforward method is to generate a large number of alignment scores by shuffling the two sequences which are to be compared and to take a histogram of this distribution. A more precise alternative is the island method (Altschul *et al.* 2001; Olsen *et al.* 1999). However, these methods are rather time consuming. Thus, the two Gumbel parameters have to be pre-computed for some few fixed scoring systems in practice (Altschul and Gish 1996; Altschul *et al.* 1997).

The necessity of pre-computing the Gumbel parameters becomes especially problematic, if adaptive schemes like, e.g., PSI-BLAST (Altschul *et al.* 1997) are being used. These schemes change their scoring system recursively depending on the sequence data they are confronted with and thus have to be able to find the two Gumbel parameters after each update of the scoring system from scratch. Another aspect which crucially depends on the real-time availability of the Gumbel parameters is the assessment of the statistical significance taking into account the amino acid composition of the individual sequences compared. Pre-computed values of the two Gumbel parameters are only correct for sequences which follow the overall amino acid frequencies which have been used in the computation of the Gumbel parameters.

In this paper, we will present a new approach to the numerical estimation of the more important of the two Gumbel parameters λ. It relies on an importance sampling technique which allows us to directly generate typical high scoring configurations instead of having to wait for them. This leads to a gain in speed of at least a factor of five in the estimation of λ. The actual runtimes on a modern workstation turn out to be in the range between under a second and a few minutes and therefore open the possibility for interactive applications.

In order to develop the algorithm we will first review the most common alignment algorithm and the significance assessment problem. In Sec. 2.3 we will put forward a conjecture on the value of the Gumbel parameter λ on which our algorithm is based and give some heuristic arguments to support this conjecture. Sec. 2.4 then describes how this conjecture can be exploited in terms of an algorithm. Since this algorithm calls for a numerical estimation of an expectation value that is dominated by *rare events* we devote the whole Sec. 3 to an importance sampling scheme which is able to perform this estimation very efficiently. In Sec. 4 we give more details on the final implementation of the whole algorithm and study its performance in comparison with the more traditional methods. Finally, we summarize our results in Sec. 5.

2 Sequence Alignment

2.1 Sequence alignment review

We want to start by giving a short review of sequence alignment. This helps us to get familiar with the question of significance assessment in sequence alignment and is an opportunity to introduce the notation we will use throughout the paper.

In database search applications the goal of sequence alignment is to assign to a given pair of biological sequences $\vec{a} = a_1 \dots a_M$ and $\vec{b} = b_1 \dots b_N$ a measure of their relative similarity. This is usually constructed from a *local* similarity matrix $s_{a,b}$ which assigns a similarity score to every pair of letters (a, b). Commonly used local similarity matrices are the PAM (Dayhoff *et al.* 1978) or BLOSUM (Henikoff and Henikoff 1992) matrices. These local similarities can be combined in various ways in order to obtain a total similarity score $\Sigma(\vec{a}, \vec{b})$.

The simplest alignment algorithm following this scheme is *gapless local* alignment. In gapless local alignment, every pair of substrings of equal length ℓ of the two sequences is assigned a similarity score

$$S(i, j, \ell) \equiv \sum_{k=0}^{\ell-1} s_{a_{i-k}, b_{j-k}}. \tag{1}$$

The total similarity score for a pair of sequences is

then given by the *best* local gapless alignment as

$$\Sigma(\vec{a},\vec{b}) \equiv \max_{i,j,\ell} S(i,j,\ell). \tag{2}$$

Although it involves an optimization over *three* independent variables, i.e., the position in the first sequence, the position in the second sequence, and the length of the alignment, it can be conveniently calculated using a dynamic programming algorithm. This dynamic programming scheme is expressed in terms of the auxiliary quantity

$$S_{i,j} \equiv \max_{\ell} S(i,j,\ell), \tag{3}$$

i.e., in terms of the maximum score $S_{i,j}$ of a local alignment *of arbitrary length* ending at the letter pair (i,j). It can be calculated in $O(NM)$ time through the recursion relation

$$S_{i,j} = \max\{S_{i-1,j-1} + s_{a_i,b_j}, 0\}. \tag{4}$$

Then, the total similarity score can be obtained in another $O(NM)$ operation

$$\Sigma(\vec{a},\vec{b}) \equiv \max_{i,j} S_{i,j}. \tag{5}$$

This algorithm has been very successfully implemented in the original version of the BLAST program (Altschul *et al.* 1990).

However, due to the possibility of insertion and deletion processes during biological evolution weak sequence homologies can only be detected if *gaps* are allowed within an alignment (Pearson 1991). These gaps can occur in both of the two sequences compared. In a given alignment each gap contributes a (negative) gap cost to the total score of the alignment. The task of an alignment algorithm *with gaps* is to find the highest scoring alignment among all the possibilities of inserting gaps in the two sequences compared. For the sake of simplicity, we will treat in our exposition only the case of a linear gap cost in which the cost assigned to a gap extending over k positions is $-\delta k$. However, everything said below extends to the more relevant case of affine gap costs where a gap of length k is assigned a cost of $-\delta - \varepsilon(k-1)$. The relevant formulae for the affine gap cost case are summarized without further derivation in the appendix and our numerical examples presented in Sec. 4 all use an affine gap cost function.

The older algorithm performing the task of finding an optimal alignment in the presence of gaps is the Needleman-Wunsch algorithm (Needleman and Wunsch 1970). It finds the highest scoring *global* alignment of two sequences. It also uses a dynamic programming scheme. The intermediary variables are the scores $H_{i,j}$ of the optimal alignment of the sequences $a_1 \ldots a_i$ and $b_1 \ldots b_j$. Since there are only three possible choices for the last position in such an optimal alignment these intermediate scores are given by the recursion equation

$$H_{i,j} = \max\{H_{i-1,j-1} + s_{a_i,b_j}, H_{i,j-1} - \delta, H_{i-1,j} - \delta\} \tag{6}$$

together with the initial conditions

$$H_{0,0} = 0, \quad H_{0,j} = -\delta j, \quad \text{and } H_{i,0} = -\delta i. \tag{7}$$

The total score of the global alignment of the two sequences $\vec{a}$ and $\vec{b}$ is then given by the final score $H_{M,N}$ of this recursion.

However, in many cases the two sequences compared are not expected to be homologous throughout their whole length. In this case, the score $\Sigma(\vec{a},\vec{b})$ of the best *local* alignment is the more interesting quantity. Analogously to gapless local alignment as discussed above, a local alignment is an alignment of two substrings $a_i \ldots a_k$ and $b_j \ldots b_l$ of the two sequences $\vec{a}$ and $\vec{b}$. For each choice of the substrings, i.e., for each choice of the positions i, j, k, and l, the score $S(i,j,k,l)$ for the optimal way of inserting gaps in an alignment of the two substrings can be calculated. The total similarity score for the two sequences $\vec{a}$ and $\vec{b}$ is then the score of the best such alignment

$$\Sigma(\vec{a},\vec{b}) \equiv \max_{i,j,k,l} S(i,j,k,l). \tag{8}$$

By a combination of the dynamic programming schemes presented above for gapless local and for global alignment with gaps, this optimization problem can be solved by the Smith-Waterman algorithm (Smith and Waterman 1981)

$$S_{i,j} = \max\{S_{i-1,j-1} + s_{a_i,b_j}, S_{i,j-1} - \delta, S_{i-1,j} - \delta, 0\} \tag{9}$$

and

$$\Sigma(\vec{a},\vec{b}) = \max_{i,j} S_{i,j} \tag{10}$$

in $O(MN)$ time. A heuristic approximation to the affine gap cost version of this algorithm is implemented, e.g., in the current versions of BLAST and PSI-BLAST (Altschul *et al.* 1997).

Depending on the choice of the scoring parameters, i.e., the matrix $s_{a,b}$ and the gap cost δ, the Smith-Waterman algorithm can be in two very distinct phases (Arratia and Waterman 1994). This result can be intuitively understood as follows: If the scoring parameters are chosen such that the intermediate score $S_{i,j}$ tends to increase with increasing i

and j when applied to a typical pair of sequences the algorithm is in the "linear" phase. In this case the "zero" alternative which distinguishes the recursion Eq. (9) from the recursion Eq. (6) will never actually be chosen. Thus, the local alignment algorithm becomes equivalent to the global alignment algorithm and will return the best *global* alignment of the two sequences. Truly local alignments are only achieved if the score $S_{i,j}$ tends to decrease with i and j. This is called the "logarithmic" phase of local alignment. In the following we will always assume that the scoring parameters are chosen such that the algorithm is in this logarithmic phase.

2.2 Significance assessment

As already mentioned in the introduction, it is not quite enough to be able to calculate the *score* of an optimal sequence alignment. In order to judge if a given score indicates a true homology between the two sequences, we have to know how probable it is that we could have obtained the same score from aligning completely unrelated, e.g., randomly chosen, sequences, i.e., we want to be able to convert the score into a p- or E-value. This is only possible if the distribution of scores which we obtain upon aligning many pairs of *random* sequences is known very precisely.

In the case of *gapless* local alignment, there is a rigorous theory which predicts this distribution (Karlin and Altschul 1990). It turns out to be a Gumbel or "extreme value" distribution (Galambos 1978; Gumbel 1958)

$$\Pr\{\Sigma < x\} = \exp[-KMNe^{-\lambda x}] \qquad (11)$$

which is characterized by the two parameters λ and K. In addition to proving that this is the correct asymptotic distribution Karlin and Altschul could also give explicit formulae which allow to calculate the two parameters λ and K for arbitrary scoring systems. λ is given by the unique positive solution of the equation

$$\mathbf{E}[e^{\lambda s}] \equiv \sum_{a,b} p_a p_b e^{\lambda s_{a,b}} = 1 \qquad (12)$$

where the p_a are the frequencies with which the amino acids a appear in the database. The formula for K is somewhat more involved but can be solved numerically very fast as well.

In the presence of gaps, there is no such theory for the score distribution. However, there is ample evidence that the distribution is still of the Gumbel form (Smith *et al.* 1985; Collins *et al.* 1988; Mott 1992; Waterman and Vingron 1994a; Waterman and Vingron 1994b; Altschul and Gish 1996; Olsen *et al.* 1999). However, the determination of the two parameters λ and K is very challenging. In few special cases there are analytical results for this problem (Mott and Tribe 1999; Bundschuh 2000; Mott 2000; Siegmund and Yakir 2000). For the practically relevant cases, the method of choice is still computer simulation, though. While the parameter K which determines the center of the distribution is relatively easy to obtain numerically, the more important parameter λ is more difficult to get a handle on. This is due to the relatively slow (exponential) falloff of the Gumbel distribution for large scores. On the other hand, one is particularly interested in assigning precise p-values to very large scores (i.e., very rare events) since in a database of 10^5 entries even a score which would occur by chance only one in 10^5 trials ($p = 10^{-5}$) is not yet significant. Thus, one is bound to aligning large numbers of random sequences in order to get a precise estimate of the parameter λ. Typical numbers are $40,000$ sequences of length 600 each (Altschul *et al.* 2001) which takes on the order of an hour on a modern workstation. This computational burden is certainly tolerable if it has to be performed only once. However, K and λ depend on the scoring system and on the amino acid probabilities. Thus, they have to be recomputed for every new scoring system used and for every sequence with an unusual amino acid frequency. The fact that this is impossible to do with current techniques within an interactive time frame is precisely the reason why BLAST offers only a certain small number of local similarity matrices and gap costs.

2.3 The central conjecture

It is our goal to design an algorithm which can estimate the Gumbel parameter λ in the presence of gaps much more rapidly than by aligning large numbers of random sequences. The algorithm which we will present below relies on a conjecture which we formulate in this section. While we do not have a proof of this conjecture, we do offer some heuristic arguments to support it. Most importantly — as we will see in Sec. 4 — the algorithm based on this conjecture yields very precise results which is the main validation for formulating our conjecture here.

2.3.1 The conjecture

Define T_N as the global alignment score $H_{N,N}$ as calculated by the recursion equation (6) applied to

two randomly chosen sequences of length N starting from the initial conditions

$$H_{i,0} = H_{0,j} = 0. \tag{13}$$

(Note that these initial conditions differ from the usual initial conditions (7) of global alignment.) Further, define λ_N as the unique positive solution of the equation

$$\begin{aligned} 1 &= \mathbf{E}[e^{\lambda_N T_N}] = \\ &= \sum_{a_1,\ldots,a_N} \sum_{b_1,\ldots,b_N} p_{a_1} \cdots p_{a_N} p_{b_1} \cdots p_{b_N} e^{\lambda_N T_N}. \end{aligned} \tag{14}$$

under the usual condition that $\mathbf{E}[T_N] < 0$, i.e., that we are in the logarithmic phase of local alignment. Then, we conjecture:

(i) The limit

$$\lambda \equiv \lim_{N\to\infty} \lambda_N \tag{15}$$

exists as a positive real number.

(ii) The distribution of the optimal scores σ of individual high scoring segments in local alignment (the "islands", see below) is asymptotically an exponential distribution with $\Pr\{\sigma > x\} \approx C^* e^{-\lambda x}$ with λ as defined in (i).

(iii) The optimal local alignment score distribution follows a Gumbel distribution $\Pr\{\Sigma < x\} = \exp[-KMNe^{-\lambda x}]$ with λ as defined in (i).

(iv) The convergence in (i) is of the form

$$\lambda_N = \lambda + \frac{c}{N} + O\left(\frac{1}{N^2}\right) \tag{16}$$

with some unknown constant c.

2.3.2 Numerical verification

The main strength of this conjecture is that points (i) and (iv) lead to a specific recipe which allows the calculation of the parameter λ of the score distribution of local alignment. As far as the validity of this conjecture is concerned, it is useful to study it at least numerically. To this end, we generate numerical estimates of λ_N for the BLOSUM62 scoring matrix and an affine gap cost of $11+k$ for a gap of length k using the method discussed in Sec. 3. As shown in Fig. 1 the estimates indeed converge as indicated in (i) and (iv) towards the known value $\lambda = 0.2670$ of the Gumbel parameter for this scoring system. A comparison to other scoring systems (data not shown) verifies that this behavior is generic, i.e., independent of the scoring system.

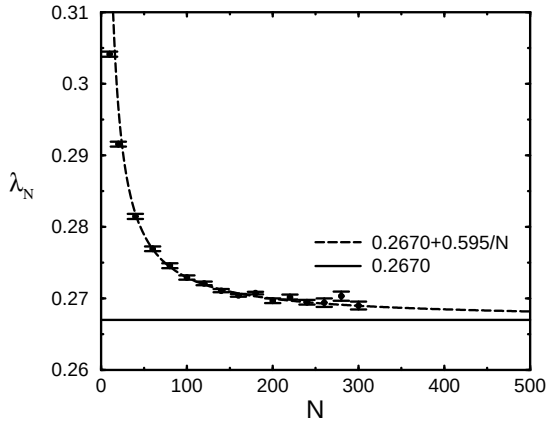

Figure 1: Estimates of approximate Gumbel parameters λ_N as a function of the global alignment size N: The scoring system used is a BLOSUM62 scoring matrix with an affine gap cost of $11+k$ for a gap of length k. The horizontal line indicates the known value of the Gumbel parameter $\lambda = 0.2670$ for this scoring system. The dashed line represents the best fit to the estimated values of λ_N to the functional form $\lambda + c/N$ in the range $60 \le N \le 180$ where the estimates for λ_N could be obtained with a precision of $\pm 0.0025\%$. In spite of the statistical fluctuations for larger N it can be seen that the estimates of λ_N indeed converge as $\lambda + c/N$ towards the limiting value of $\lambda = 0.2670$.

While we will discuss the algorithm following from the conjecture in Sec. 2.4 we will now give some additional arguments for the validity of our conjecture. Therefore, the reader only interested in the algorithmic application can skip directly to Sec. 2.4. These arguments are in no means meant to be a replacement for a full proof of the central conjecture. We will mainly take the point of view, that the central conjecture suggests the algorithm presented in this paper. This algorithm turns out to work reasonably well for our practically relevant examples which makes it useful independently of the fact if the central conjecture is proven or not.

2.3.3 The gapless case

First of all, it is reassuring to note that the conjecture is correct in the case of gapless alignment (i.e., in the case of infinite gap cost.) In the gapless case, we simply have $T_N = \sum_{i=1}^{N} s_{a_i,b_i}$. Thus, condition Eq. (14) becomes $1 = \mathbf{E}[e^{\lambda_N T_N}] = (\mathbf{E}[e^{\lambda_N s}])^N$. Thus, all λ_N are given by the N-independent condition $\mathbf{E}[e^{\lambda_N s}] = 1$ which we recognize as the condition Eq. (12) of the gapless case. Therefore, (i) and (iv)

are trivially valid with $c = 0$ and (ii) and (iii) reduce to the result of Karlin and Altschul (1990).

2.3.4 Heuristic arguments

Now we want to give some very heuristic arguments for the validity of the full central conjecture. It is largely a review of some ideas on the origin of the Gumbel distribution which have been pointed out earlier (see, e.g., (Olsen *et al.* 1999).)

The main observation which leads to the appearance of the Gumbel distribution is that the recursion Equation (9) of the Smith-Waterman algorithm contains three different cases. (i) the score $S_{i,j}$ can be zero. If the score $S_{i,j}$ is not zero, one of the first three alternatives in the maximum of Eq. (9) must be responsible for this. Thus, the positive score can be (ii) attributed to another positive score $S_{i-1,j-1}$, $S_{i,j-1}$, or $S_{i-1,j}$ or (iii) is due to the fact that $S_{i-1,j-1} = 0$ but $s_{a_i,b_j} > 0$. The last event we will call an "island initiation event". We can assign every pair (i, j) with a positive score $S_{i,j}$ to one of these island initiation events, since such a pair is either an initiation event by itself or is related to another score $S_{i,j}$ according to case (ii) above. This subdivides the pairs (i, j) with positive score $S_{i,j}$ into a collection of "islands" (an island consisting of all those pairs which are assigned to the same island initiation event) in a "sea" of zero scores. This relatively abstract definition of the islands becomes immediately obvious when plotting the score $S_{i,j}$ as a function of the "coordinates" i and j as it is done in Fig. 2.

Each of these islands I can be assigned its peak score $\sigma_I \equiv \max_{(i,j)\in I} S_{i,j}$. Assuming that the Poisson clumping heuristic (Aldous 1989) is applicable to these island peak scores, the island peak scores become asymptotically statistically independent random variables with an exponential distribution

$$\Pr\{\sigma > x\} \approx C^* e^{-\lambda x}. \tag{17}$$

This fact has also been extensively verified numerically (Olsen *et al.* 1999; Bundschuh 2000; Altschul *et al.* 2001). Then, the total score of an alignment is simply given as the maximum of these island peak scores

$$\Sigma(\vec{a}, \vec{b}) = \max_I \sigma_I. \tag{18}$$

Since the density ρ of islands on the scoring lattice is more or less constant, this maximum is taken over ρMN islands. Thus, the distribution of the local alignment score Σ is

$$\begin{aligned} \Pr\{\Sigma < x\} &= (\Pr\{\sigma < x\})^{\rho MN} \approx (1 - C^* e^{-\lambda x})^{\rho MN} \\ &\approx \exp\left[-C^* \rho MN e^{-\lambda x}\right]. \end{aligned} \tag{19}$$

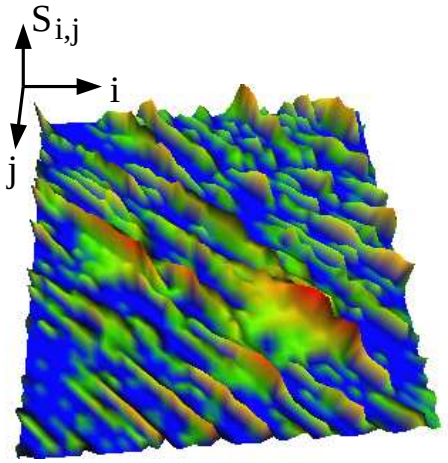

Figure 2: Scoring landscape for a local alignment of two random amino acid sequences using the BLOSUM62 scoring matrix and a gap cost of $11 + k$ for a gap of length k: The figure shows the score $S_{i,j}$ as a function of the positions in the sequences i and j. It can be clearly seen how the scoring landscape is decomposed into several "islands" separated by a "sea" of zero scores.

From this argument we do not only see how the extreme value distribution occurs very naturally but also that the parameter λ which we are interested in appears already in the island peak score distribution Eq. (17). Thus, it is enough to study the island peak score distribution in order to determine the parameter λ. Specifically, we have to argue that the parameter λ in the island peak score distribution Eq. (17) is given by the limiting process described in (i) and (iv).

Since the scores within an island are by definition positive, an island can be thought of as a *global* alignment of two substrings of the total sequences. Thus, the global alignment score T_N obeys the same distribution as the island peak scores σ of islands for which the longer of the two substrings has length N. This is guaranteed by the specific choice of the initial conditions Eq. (13). Given, that we know that the island peak score distribution follows the exponential form given in Eq. (17) what can we say about the behavior of $\mathbf{E}[e^{xT_N}]$? Because we are in the local alignment regime where global alignment scores are typically *negative*, $e^{xT_N} \approx 0$ for *most* pairs of sequences. However, we know that in a small fraction of cases — the fraction being proportional to $e^{-\lambda\sigma}$ — the score is at least $T_N \geq \sigma$. These scores contribute with at least $e^{x\sigma}$ to $\mathbf{E}[e^{xT_N}]$. If we look at $\mathbf{E}[e^{xT_N}]$ as a function of N, there are two dif-

ferent scenarios. If $x < \lambda$, the islands with reasonably large scores become rarer at a faster pace than the factor e^{xT_N} can compensate for and $\mathbf{E}[e^{xT_N}]$ will converge to zero. If on the other hand $x > \lambda$ the factor e^{xT_N} over-compensates for the rareness of these large scores and $\mathbf{E}[e^{xT_N}]$ diverges with N. Exactly at $x = \lambda$ there is a balance between the rareness of islands with a large score and the amplification of these scores by the factor e^{xT_N} for *all* N. Thus, by fixing $\mathbf{E}[e^{\lambda_N T_N}] = 1$ we demand that λ_N be close enough to λ to neither have converged towards zero nor towards infinity yet. Upon increasing N, λ_N thus has to converge towards the λ which describes the island peak score distribution and therefore also the distribution of the total alignment scores.

2.3.5 A partial proof

Finally, parts of a slightly modified version of the conjecture can actually be *proven* as a consequence of a study by Zhang (1995). The necessary modification is the following: Let us denote by $\widehat{T}_N$ the fully global alignment score of two random sequences of length N, i.e., $\widehat{T}_N \equiv H_{N,N}$ where $H_{i,j}$ is calculated according to recursion Eq. (6) and the initial conditions Eq. (7). With this quantity we can again define $\widehat{\lambda}_N$ as the unique solution of $\mathbf{E}[e^{\widehat{\lambda}_N \widehat{T}_N}] = 1$ (assuming $\mathbf{E}[\widehat{T}_N] < 0$.) The difference between the $\widehat{\lambda}_N$ and the λ_N used in the central conjecture is the different initial condition for the $H_{i,j}$. The quantity $\widehat{T}_N$ is the usual global alignment score of two sequences while the quantity T_N used in the central conjecture is a global alignment score that does not penalize initial gaps. Since the number of initial gaps of an optimal alignment of two random sequences typically grows much less than linear in the length N of the sequences, we expect that using λ_N or $\widehat{\lambda}_N$ in the central conjecture is equivalent as N becomes large. The reason why we chose λ_N over $\widehat{\lambda}_N$ is that the λ_N numerically seem to approach the asymptotic large N behavior *faster* than the $\widehat{\lambda}_N$ which makes a difference in the practical implementation of the algorithm.

We will now argue how statement (i) and parts of statement (iii) of the central conjecture if formulated in terms of the $\widehat{\lambda}_N$ follow from (Zhang 1995). If we explicitly denote the local alignment score of two random sequences of length M by $\Sigma(M)$, we can rewrite Eqs. (2.15) and (2.16) of (Zhang 1995) in our notation. They imply that given $\varepsilon > 0$ and large enough N and n the inequality

$$\frac{\Sigma(Nn)}{\log n} + \varepsilon \geq \frac{2}{\widehat{\lambda}_N} \geq 2(b - \varepsilon)\left(1 - \frac{\varepsilon}{r(0)}\right) \qquad (20)$$

holds where b and $r(0)$ are positive constants independent of ε, n, and N. According to Theorem 1 of (Zhang 1995) we get in the limit $n \to \infty$ almost surely

$$2b + \varepsilon \geq \frac{2}{\widehat{\lambda}_N} \geq 2(b - \varepsilon)\left(1 - \frac{\varepsilon}{r(0)}\right). \qquad (21)$$

This implies that $\lim_{N\to\infty} \widehat{\lambda}_N = 1/b$ exists as a positive real number, i.e., part (i) of the central conjecture is true for the $\widehat{\lambda}_N$. Moreover, if we believe the numerical evidence (Smith *et al.* 1985; Collins *et al.* 1988; Mott 1992; Waterman and Vingron 1994a; Waterman and Vingron 1994b; Altschul and Gish 1996; Olsen *et al.* 1999) and *assume* that the distribution of the local alignment score $\Sigma(M)$ is of the Gumbel form with some Gumbel parameter λ, it is easy to see that $\lim_{M\to\infty} \Sigma(M)/\log M = 2/\lambda$. Theorem 1 of (Zhang 1995) says that this limit equals $2b$ almost surely. Combining this with our derivation of (i) yields that the Gumbel parameter λ of local alignment is indeed given by the limiting process $\lambda = \lim_{N\to\infty} \widehat{\lambda}_N$, i.e., (iii) holds under the assumption that the form of the distribution is Gumbel.

2.4 Implementation

The central conjecture leads to a very simple way to calculate λ numerically if we are able to estimate the global alignment expectation value $\mathbf{E}[e^{xT_N}]$ properly. While we devote the whole Sec. 3 to the question of how to obtain this expectation value, we want to outline the rest of the algorithm already now. According to the central conjecture we simply proceed as follows

1. Estimate $\mathbf{E}[e^{xT_N}]$ for $N = 60$, $N = 80$, and $N = 100$.
2. Solve for the unique positive solutions λ_{60}, λ_{80}, and λ_{100} of the equations $\mathbf{E}[e^{xT_N}] = 1$ for $N = 60$, $N = 80$, and $N = 100$ respectively.
3. Fit the functional form $\lambda_N = \lambda + c/N$ to the three values $N = 60$, $N = 80$, and $N = 100$ using λ and c as the two fitting parameters.

This yields the Gumbel parameter λ of the corresponding local alignment algorithm as one of the fitting parameters in step 3.

It is important to note that the traditional techniques to estimate λ require pairs of *long* random sequences to be aligned to each other in order to obtain good estimates. Typical lengths at which the estimates become reliable are $N = 600$ amino acids or more. This is due to the known influence of

the sequence length on the statistics of local alignment (Altschul and Gish 1996; Altschul *et al.* 2001). Here, each global alignment corresponds to a high scoring path or island by itself. Typical high scoring paths in alignments of random sequences are only of the order of a few tens of amino acids long — if the scoring system and gap costs were chosen such that high scoring paths of random alignments were longer than that real biological homologies which typically come in lengths of no more than 100 amino acids would not "stand out" of this background any more and would not be detectable. Thus, for practically useful scoring systems and gap costs, high scoring paths of lengths $N = 60$. $N = 80$ and $N = 100$ have to be considered *very long*. This reduction in the length of the sequences to be aligned compared to the traditional methods is the main computational advantage of our procedure. It makes a rather large difference since the computational complexity of sequence alignment of two sequences of length N is $O(N^2)$.

3 Importance sampling

Obviously, the algorithm presented in the last section crucially depends on our ability to obtain the expectation value $\mathbf{E}[e^{xT_N}]$ in an effective way. The most straightforward way which comes to mind is to draw a large number $\mathcal{N}$ of sequence pairs $(\vec{a}_i, \vec{b}_i)$ of length N in which each letter is chosen independently according to the amino acid frequencies p_a. Then, we can calculate the global alignment score $T_N(\vec{a}_i, \vec{b}_i)$ using the recursion Eq. (6) and the initial conditions Eq. (13) and finally estimate

$$\mathbf{E}[e^{xT_N}] \approx \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} e^{xT(\vec{a}_i, \vec{b}_i)}. \tag{22}$$

However, the *typical* global alignment scores T_N are negative and hardly contribute to the expectation value in question. The main contribution to the expectation value stems from the *rare events* with a positive global alignment score. Thus, we would have to choose an extremely large number $\mathcal{N}$ of sequences in order to include enough of these *rare* events to get a good estimate of $\mathbf{E}[e^{xT_N}]$.

Therefore, we will take advantage of an *importance sampling* technique. Instead of randomly drawing many sequence pairs which in the end do not contribute to the average anyways, we will bias the generation of random sequence pairs in such a way that *all* pairs which we align contribute to the expectation value $\mathbf{E}[e^{xT_N}]$. Of course, we have to correct for the bias in the distribution from which we draw the sequence pairs appropriately. If $P_{\text{biased}}(\vec{a}, \vec{b})$ is the probability to draw the sequence pair $(\vec{a}, \vec{b})$ in the appropriately biased ensemble, then such a sequence pair is overrepresented by a factor of

$$W(\vec{a}, \vec{b}) = \frac{P_{\text{biased}}(\vec{a}, \vec{b})}{\prod_{i=1}^{N} p_{a_i} p_{b_i}}. \tag{23}$$

Thus, the estimate of the expectation value we are interested in can be obtained in the following way:

1. Draw $\mathcal{N}$ sequence pairs $(\vec{a}_i, \vec{b}_i)$ from the appropriately biased ensemble of sequence pairs which contribute significantly to the expectation value $\mathbf{E}[e^{xT_N}]$.

2. Calculate the global alignment scores $T_N(\vec{a}_i, \vec{b}_i)$ using the recursion Eq. (6) and the initial conditions Eq. (13) for all the chosen sequence pairs.

3. Calculate the relative probabilities $W(\vec{a}_i, \vec{b}_i)$ for all chosen sequence pairs.

4. Estimate

$$\mathbf{E}[e^{xT_N}] \approx \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} e^{xT(\vec{a}_i, \vec{b}_i)} [W(\vec{a}_i, \vec{b}_i)]^{-1}. \tag{24}$$

This is a rapid and reliable method to estimate $\mathbf{E}[e^{xT_N}]$ if we are able to choose sequence pairs which are indeed representative of the major contributions to $\mathbf{E}[e^{xT_N}]$ (for x close to λ_N) and if we are able to calculate the relative probabilities $W(\vec{a}, \vec{b})$ in an effective way. These two points will be discussed in the following in more detail.

3.1 Choosing sequence pairs

As already said, the sequence pairs have to be chosen such that they are as representative as possible of the major contributions to $\mathbf{E}[e^{xT_N}]$ (for x close to λ_N). Recalling the origin of this expectation value shows that we have to choose sequence pairs as they *typically* arise *within very high scoring paths* of local sequence alignment. Fortunately, there exists a variation of the Smith-Waterman algorithm (the hybrid alignment algorithm (Yu and Hwa 2001)) for which the distribution of typical sequence pairs on high scoring paths is known. Here, we will simply assume that the difference between the original Smith-Waterman algorithm and the hybrid alignment algorithm is not too severe and choose our sequence pairs as typical for high scoring paths in the hybrid algorithm. Note, that this is *not* an approximation. We use the hybrid algorithm merely as an inspiration on how to choose the sequence pairs. If the sequence

pairs contributing to $\mathbf{E}[e^{xT_N}]$ are notably different from typical sequence pairs of high scoring paths of the hybrid algorithm, we merely lose computational efficiency in the estimation of $\mathbf{E}[e^{xT_N}]$. We will notice this by monitoring the statistical deviations of our estimates. Thus, we will have to choose the larger numbers $\mathcal{N}$ of sequence pairs the larger the discrepancy between the Smith-Waterman and the hybrid high scoring paths but we will not obtain wrong results if such a discrepancy occurs.

Since the way of generating sequence pairs which are typical of high scoring paths is described in detail in (Yu and Hwa 2001), we will here only summarize the results in the case of linear gap cost. The appendix shows the more complicated case of an affine gap cost. The main point is that the two sequences generated are *correlated* according to the following construction scheme:

1. Start with empty sequences $\vec{a}$ and $\vec{b}$
2. Repeat until one of the two sequences reaches length N:
   - With probability μ choose a random amino acid a according to the background amino acid frequencies p_a and append it to sequence $\vec{a}$
   - With probability μ choose a random amino acid b according to the background amino acid frequencies p_b and append it to sequence $\vec{b}$
   - With probability $(1-2\mu)$ choose a pair (a,b) of amino acids according to some joint distribution $q_{a,b}$ and append a to $\vec{a}$ and b to $\vec{b}$
3. Generate as many random amino acids a according to the background amino acid frequencies p_a and append them to the shorter of the two sequences as necessary to make the length of the shorter sequence N as well.

Obviously, the correct choice of the probabilities μ and $q_{a,b}$ has to depend on the scoring matrix $s_{a,b}$ and the gap cost δ. According to Yu and Hwa (2001) these probabilities have to be chosen as follows: Define λ_0 as the Gumbel parameter of the given scoring matrix *in the absence* of gaps, i.e., as the unique positive solution of Eq. (12). Then,

$$q_{a,b} \equiv p_a p_b e^{\lambda_0 s_{a,b}} \qquad \text{and} \qquad \mu \equiv e^{-\lambda_0 \delta} \tag{25}$$

are the probabilities that lead to sequence pairs which are typical for the high scoring paths of hybrid alignment. Note, that due to the definition of λ_0 the $q_{a,b}$ are indeed a (normalized) probability distribution for pairs (a,b) of amino acids and that this is the same distribution which is known to generate typical high scoring paths in *gapless* alignment (Karlin and Altschul 1990).

3.2 Computing the relative probability

The last obstacle in the implementation of our algorithm is the calculation of the relative probabilities $W(\vec{a},\vec{b})$ defined in Eq. (23). They measure how probable it is to choose a given sequence pair $(\vec{a},\vec{b})$ by using the algorithm given in Sec. 3.1 (with given values of $q_{a,b}$ and μ) relative to just choosing the sequences letter by letter with the background probabilities p_a. This problem has already been solved by Bishop and Thompson (1986) in the case of a linear gap cost; it has been generalized to affine gap costs by Yu and Hwa (2001) (see also the Appendix.) Basically, the relative probability can be calculated in the same way as the global alignment score T_N. The intermediary quantity is in this case the relative probability $W_{i,j}$ to generate the sequence pair $(a_1 \ldots a_i, b_1 \ldots b_j)$. This sequence pair could have been generated in three different ways. The last step could have been the addition of the letter a_i which happens with a probability μp_{a_i}. It also could have been the addition of the letter b_j which happens with a probability μp_{b_j} or the addition of a letter pair with probability $(1-2\mu)q_{a_i,b_j}$. The probabilities of these three cases have to be added up in order to calculate the probability for the sequence pair. Dividing by the background probabilities of the same events leads to the recursion equation

$$W_{i,j} = \mu W_{i,j-1} + \mu W_{i-1,j} + (1-2\mu)\frac{q_{a_i,b_j}}{p_{a_i}p_{b_j}} W_{i-1,j-1} \tag{26}$$

for the relative probabilities $W_{i,j}$. The appropriate initial conditions are

$$W_{i,0} = \mu^i \quad \text{and} \quad W_{0,j} = \mu^j. \tag{27}$$

Finally, according to step 3 of the algorithm presented in Sec. 3.1 the total probability for choosing a sequence pair $(\vec{a},\vec{b})$ is given by

$$\begin{aligned} W(\vec{a},\vec{b}) = &\sum_{i=1}^{N-1}\left[\mu + (1-2\mu)\frac{q_{a_i,b_N}}{p_{a_i}p_{b_N}}\right] W_{i-1,N-1} + \\ &+ \sum_{j=1}^{N}\left[\mu + (1-2\mu)\frac{q_{a_N,b_j}}{p_{a_N}p_{b_j}}\right] W_{N-1,j-1} + \\ &+ \mu W_{N-1,N-1}. \end{aligned} \tag{28}$$

Thus, the relative probability $W(\vec{a},\vec{b})$ for two sequences of length N can be conveniently calculated in $O(N^2)$ time.

4 Performance

In this section we want to discuss the actual implementation of the algorithm which we presented above and study its performance in a realistic setting.

4.1 Implementation details

We implemented the complete algorithm in the more general case of affine gap costs. The source code can be obtained upon request from the author.

One point of consideration is the statistical error in the value of λ_N. It is estimated by keeping track of the standard deviation of the estimate for $\mathbf{E}[e^{xT_N}]$ and calculating the worst-case values of λ_N if the estimate for $\mathbf{E}[e^{xT_N}]$ is allowed to vary within one standard deviation. The algorithm then dynamically chooses the number $\mathcal{N}$ of sequence pairs such that the relative statistical error on the individual values λ_{60}, λ_{80}, and λ_{100} is below $\Delta/2$ where Δ is a user-supplied requested relative precision for the final λ. This precision on the individual λ_N turns out to be sufficient to obtain the requested relative statistical error on the final value of λ.

Another important point is that the computation time can be reduced significantly by noting that the sequence pairs are correlated. Thus, the highest scoring path in the global alignment leading to T_N must more or less follow the pattern of insertion and deletion events which took place during the *generation* of the two sequences. This is also the region which gives the largest contribution to the relative probability $W(\vec{a},\vec{b})$. Thus, the recursions Eqs. (6) and (26) only have to be performed in some band around the pattern of known insertion and deletion events for every sequence as illustrated in Fig. 3. In our current implementation, we take advantage of this fact only in a very crude way by restricting the recursions to a band of some width d around the line $i = j$. The width d is determined by the total number of insertion and deletion events which occurred during the generation of the sequence pair as indicated by the arrow in Fig. 3. Following the insertion/deletion pattern more closely can clearly lead to an even higher speed advantage than our current implementation.

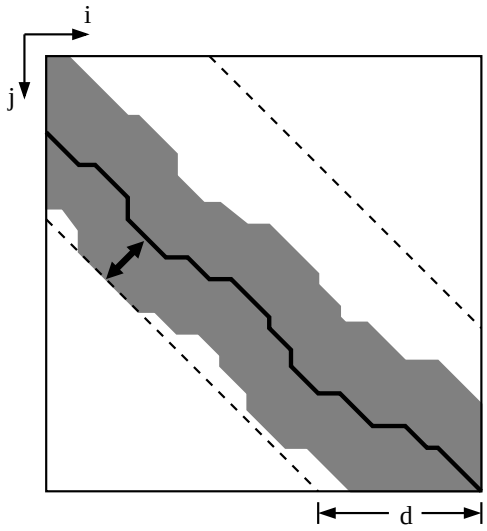

Figure 3: Regions to which the application of the recursion Eqs. (6) and (26) can be restricted: The two sequences aligned are correlated and have a known pattern of insertions and deletions according to the sequence pair generation algorithm presented in Sec. 3.1. This pattern is indicated as the solid line. Due to this sequence correlation it is known *a priori* that the highest scores appear in a small region around this known insertion/deletion pattern as indicated by the gray area. Thus, the recursion equations only have to be evaluated within this area. Our current implementation does not take full advantage of this fact but instead restricts the alignments to the region indicated by the dashed lines with the width of the band of a width d given by the requirement that the insertion/deletion pattern has to be at least 20 amino acid positions away from the boundary (as indicated by the arrow.)

4.2 Precision

We tested the validity of our algorithm on five of the scoring matrices and gap costs recommended by BLAST. We determined reference values of λ for each of the five scoring systems by the island method (Olsen *et al.* 1999; Altschul *et al.* 2001). We also used our algorithm to estimate λ in all of the five cases. As the required relative precision we used the typical values $\Delta = 4\%$ and $\Delta = 0.5\%$ (Note, that the actual p-values depend very sensitively on the estimated value of λ such that an error in the estimation of λ by 1% can already lead to an error in p-values by a factor of two in the vicinity of the significance threshold of a typical database. Thus, it is crucial to obtain the estimate of λ with a very high precision.) The different estimates of λ are

Scoring system	$\lambda_{reference}$	$\Delta = 0.5\%$			$\Delta = 4\%$		
		λ	DP steps	time (m:s)	λ	DP steps	time (sec)
BLOSUM45/14/2	0.1961 ± 0.0008	0.1978 (0.9%)	$528 \cdot 10^6$	3:00	0.1991 (1.5%)	$4.2 \cdot 10^6$	3.0
BLOSUM62/11/1	0.2670 ± 0.0002	0.2669 (0.04%)	$1126 \cdot 10^6$	5:49	0.2748 (2.9%)	$4.3 \cdot 10^6$	2.5
BLOSUM80/10/1	0.2993 ± 0.0012	0.3004 (0.4%)	$210 \cdot 10^6$	1:00	0.3053 (2.0%)	$1.0 \cdot 10^6$	0.6
PAM70/10/1	0.2921 ± 0.0013	0.2922 (0.03%)	$106 \cdot 10^6$	0:26	0.2941 (0.7%)	$1.0 \cdot 10^6$	0.6
PAM30/9/1	0.2963 ± 0.0011	0.2954 (0.3%)	$113 \cdot 10^6$	0:27	0.2967 (0.1%)	$1.1 \cdot 10^6$	0.6

Table 1: Estimates of the Gumbel parameter λ for various scoring matrices and gap costs. The scoring parameters are given in the form Matrix/$(\delta+\varepsilon)/\varepsilon$ where a gap of length k is assigned a cost of $-\delta-(k-1)\varepsilon$. The second column gives the reference values as generated by the island method (Olsen *et al.* 1999; Altschul *et al.* 2001). The other columns are the results of our new algorithm at a requested precision of $\Delta = 0.5\%$ and $\Delta = 4\%$. As the actual deviations from the reference values given in the parentheses show, the estimates are within the required precision — most of them actually well within — of the reference value. The computational effort necessary is measured in two ways. On the one hand the number of dynamic programming steps performed is shown. It has to be compared to $6,400 \cdot 10^6$ for $\Delta = 0.5\%$ and $40 \cdot 10^6$ for $\Delta = 4\%$ for the traditional methods of estimating λ. Thus, our algorithm is at least by a factor of 6 faster than the traditional methods. The absolute execution times on a 600MHz Pentium III processor show that the estimation of λ is moved into the realm of interactive applications by our algorithm.

shown in Table 1. The values of λ estimated by our algorithm are within the requested precision of the reference values — specifically if one takes into account the fact that the reference values are also only known up to a precision of 0.5%. Thus, we conclude that our algorithm provides *accurate* estimates of λ in spite of the heuristic steps taken in its theoretical development.

4.3 Computational effort

The computational effort is best measured by the number of dynamic programming steps as given by Eqs. (6) and (26) which have to be performed. Due to the dynamic choice of the number of sequences and of the band which is actually scored as described in Sec. 4.1 this number is different for different scoring systems. These numbers are also shown in Table 1 for all five scoring systems and for both relative precisions. In order to estimate λ to a precision of 0.5% using traditional techniques, typically on the order of $40,000$ pairs of sequences of length 600 have to be aligned. This corresponds to roughly $6,400 \cdot 10^6$ dynamic programming steps. Thus, even in the worst of the five cases, our new algorithm is by a factor of six faster. Under the more relaxed conditions of 4% precision, the traditional methods still need about $40 \cdot 10^6$ dynamic programming steps (Altschul *et al.* 2001). In this regime, our new method is faster by a factor of at least 10. In order to illustrate the practicability of our new method, Table 1 also contains the actual computation times needed for the estimates on a 600MHz Pentium III processor. At a required precision of $\Delta = 4\%$ they are clearly within the range of interactive applications and even at a required precision of $\Delta = 0.5\%$ the computation time is still very reasonable.

5 Discussion

In this paper we presented a new method to estimate the important Gumbel parameter λ of local sequence alignment with gaps. We verified its performance on some typical scoring systems. It correctly estimates the values of λ in all cases. In spite of a less than optimal implementation our new algorithm is faster than the traditional methods of estimating λ by at least a factor of 6. An actual runtime in the range of below a second to a few minutes on a modern workstation brings the estimation of λ into the realm of interactive applications. Thus, this algorithm can, e.g., be used in order to choose optimized scoring systems for high quality pairwise alignments in an automatic, iterative fashion.

6 Acknowledgments

We gratefully acknowledge many enlightening discussions with S. Altschul, A. Dembo, T. Hwa, R. Olsen, and Y.-K. Yu. This work has been supported in part by the National Science Foundation through Grants No. DMR-9971456 and DBI-9970199.

Appendix: affine gap cost

For completeness, we summarize in this appendix the formulae necessary to apply our method to affine gap costs, where a cost of $-\delta-(k-1)\varepsilon$ is associated with every gap of length k. They have been omitted in the main text for the sake of clarity of the presentation.

The necessary modification of the recursion equation (6) for global alignment is well-known and can for example be found in (Waterman 1995). Instead of the intermediate variable $H_{i,j}$ we now have to use three intermediate variables $H^S_{i,j}$, $H^I_{i,j}$ and $H^D_{i,j}$ which obey the recursion equations

$$H^S_{i,j} = \max\{H^S_{i-1,j-1}, H^I_{i-1,j-1}, H^D_{i-1,j-1}\} + s_{a_i,b_j} \quad (29)$$

$$H^D_{i,j} = \max\{H^S_{i-1,j} - \delta, H^D_{i-1,j} - \epsilon\} \quad (30)$$

$$H^I_{i,j} = \max\{H^S_{i,j-1} - \delta, H^I_{i,j-1} - \epsilon, H^D_{i,j-1} - \delta\}. \quad (31)$$

The usual global alignment score of two sequences of length M and N respectively, is calculated as

$$H_{M,N} = \max\{H^S_{M,N}, H^D_{M,N}, H^I_{M,N}\} \quad (32)$$

after applying the recursion equations (29)-(31) to the initial conditions

$$H^S_{0,0} = H^D_{0,0} = H^I_{0,0} = H^S_{i,0} = H^S_{0,j} = H^D_{0,j} = H^I_{i,0} = 0,$$
$$H^D_{i,0} = -\delta-(i-1)\epsilon \quad \text{and} \quad H^I_{0,j} = -\delta-(j-1)\epsilon \quad (33)$$

for $i \in \{1,\ldots,M\}$ and $j \in \{1,\ldots,N\}$. For our method we want to calculate the semi-global alignment score T_N of a pair of sequences of length N. As in the case of linear gap cost it differs from the global alignment score $H_{N,N}$ only in so far as initial gaps are not penalized. Thus, it is calculated as

$$T_N = \max\{H^S_{N,N}, H^D_{N,N}, H^I_{N,N}\} \quad (34)$$

after applying the recursion equations (29)-(31) to the modified initial conditions

$$H^S_{i,0} = H^D_{i,0} = H^I_{i,0} = H^S_{0,j} = H^D_{0,j} = H^I_{0,j} = 0 \quad (35)$$

for $i,j \in \{0,\ldots,N\}$.

In the presence of an affine gap cost, obviously the sequence pairs which contribute most to the expectation value $\mathbf{E}[e^{xT_N}]$ will be different from the relevant sequence pairs in the linear gap cost case presented in Sec. 3.1. However, the alternative alignment algorithm by Yu and Hwa (2001) which provides us with the intuition on how to choose good sequence pairs applies to affine gap costs as well. Thus, we can take the recipe for generating sequence pairs from Yu and Hwa (2001) again assuming that sequence pairs which are relevant for the alternative alignment algorithm are not too different from sequence pairs which contribute most to $\mathbf{E}[e^{xT_N}]$.

The procedure of generating sequence pairs given in Sec. 3.1 can be cast in terms of a hidden Markov model with three states, namely, substitution, insertion and deletion. In the presence of affine gap costs, this becomes more obvious since now the transition probabilities depend on the current state while in the case of linear gap costs the new state could be chosen with fixed probabilities independent of the current state. The affine gap cost version of the sequence pair generation algorithm then reads:

1. Start with empty sequences $\vec{a}$ and $\vec{b}$. Choose the substitution state with probability η, the insertion state with probability μ^{IS}, or the deletion state with the remaining probability μ^{DS}.

2. Repeat until one of the two sequences reaches length N:

   - If we are in the deletion state choose a random amino acid a according to the background amino acid frequencies p_a and append it to sequence $\vec{a}$. Then, stay in the deletion state with probability ν, choose the substitution state with probability μ^{SD}, or choose the insertion state with the remaining probability μ^{ID}.
   - If we are in the insertion state choose a random amino acid b according to the background amino acid frequencies p_b and append it to sequence $\vec{b}$. Then, stay in the insertion state with probability ν or choose the substitution state with the remaining probability μ^{SI}.
   - If we are in the substitution state choose a pair (a,b) of amino acids according to some joint distribution $q_{a,b}$ and append a to $\vec{a}$ and b to $\vec{b}$. Then, stay in the substitution state with probability η, choose the deletion state with probability μ^{DS}, or choose the insertion state with the remaining probability μ^{IS}.

3. Generate as many random amino acids a according to the background amino acid frequencies p_a and append them to the shorter of the two sequences as necessary to make the length of the shorter sequence N as well.

Again, this has to be supplemented by a choice of the probabilities η, μ^{IS}, μ^{DS}, μ^{SI}, μ^{SD}, μ^{ID}, ν and $q_{a,b}$ which depends on the scoring matrix $s_{a,b}$ and

the gap costs δ and ε. The correct choice according to Yu and Hwa (2001) again involves the Gumbel parameter λ_0 of the given scoring matrix in the absence of gaps, i.e., the unique positive solution of Eq. (12). With this λ_0, we can define

$$q_{a,b} \equiv p_a p_b e^{\lambda_0 s_{a,b}}, \quad \mu \equiv e^{-\lambda_0 \delta} \quad \text{and} \quad \nu \equiv e^{-\lambda_0 \varepsilon}. \tag{36}$$

The remaining transition probabilities are then given by

$$\begin{aligned} \eta &= \frac{(1-\nu)^2}{(1+\mu-\nu)^2}, & \mu^{SI} &= 1-\nu, \\ \mu^{IS} &= \frac{\mu(1-\nu)}{(1+\mu-\nu)^2}, & \mu^{DS} &= \frac{\mu}{1+\mu-\nu}, \\ \mu^{SD} &= \frac{(1-\nu)^2}{1+\mu-\nu}, \text{ and} & \mu^{ID} &= \frac{\mu(1-\nu)}{1+\mu-\nu} \end{aligned} \tag{37}$$

which can be verified to fulfill the probability conservation conditions $\nu+\mu^{SD}+\mu^{ID}=1$, $\nu+\mu^{SI}=1$, and $\eta+\mu^{IS}+\mu^{DS}=1$.

Finally, we have to be able to calculate the relative probability $W(\vec{a},\vec{b})$ of a sequence pair being generated by the above hidden Markov model. Generalizing Bishop and Thompson's (1986) approach this relative probability can be calculated with the help of the recursion equations (Yu and Hwa 2001)

$$\begin{aligned} W^S_{i,j} &= e^{\lambda_0 s_{a_i,b_j}} \times \\ &\quad \times \left[\eta W^S_{i-1,j-1} + \mu^{SI} W^I_{i-1,j-1} + \mu^{SD} W^D_{i-1,j-1}\right] \\ W^D_{i,j} &= \mu^{DS} W^S_{i-1,j} + \nu W^D_{i-1,j} \quad \text{and} \\ W^I_{i,j} &= \mu^{IS} W^S_{i,j-1} + \nu W^I_{i,j-1} + \mu^{ID} W^D_{i,j-1}. \end{aligned} \tag{38}$$

They have to be applied to the given sequence pair with the initial conditions

$$\begin{aligned} W^S_{0,0} &= 1, & W^D_{0,0} &= W^I_{0,0} = 0, \\ W^S_{0,j} &= W^S_{i,0} = W^D_{0,j} = W^I_{i,0} = 0, \\ W^D_{i,0} &= \mu^{DS}\nu^{i-1}, \text{ and} & W^I_{0,j} &= \mu^{IS}\nu^{j-1} \end{aligned} \tag{39}$$

for $i,j \in \{1,\ldots,N\}$. Then, the relative probability $W(\vec{a},\vec{b})$ of the sequence pair $(\vec{a},\vec{b})$ is given by

$$\begin{aligned} W(\vec{a},\vec{b}) \;=\; & W^S_{N,N} + W^I_{0,N} + W^D_{N,0} + \\ & + \sum_{i=1}^{N-1} \left[W^S_{i,N} + W^I_{i,N} + W^S_{N,i} + W^D_{N,i}\right]. \end{aligned} \tag{40}$$

References

Aldous, D.J. 1989. *Probability approximations via the Poisson clumping heuristic.* Springer Verlag, New York, NY.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., and Lipman, D.J. 1990. Basic Local Alignment Search Tool. *J. Mol. Biol.* 215(3), 403–410.

Altschul, S.F. and Gish, W. 1996. Local Alignment Statistics. *Methods in Enzymology* 266, 460–480.

Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D.J. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* 25(17), 3389–3402.

Altschul, S.F., Bundschuh, R., Olsen, R., and Hwa, T. 2001. The Estimation of Statistical Parameters for Local Alignment Score Distributions. *Nucleic Acids Research* 29(2).

Arratia, R., and Waterman, M.S. 1994. A phase transition for the score in matching random sequences allowing deletions. *Ann. Appl. Probab.* 4(2), 200–225.

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

Bundschuh, R. 2000. An Analytic Approach to Significance Assessment in Local Sequence Alignment with Gaps, 86–95. In Istrail, S., *et al.* eds., *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, ACM press, New York, NY.

Collins, J.F., Coulson, A.F.W., and Lyall, A. 1988. The significance of protein sequence similarities. *CABIOS* 4(1), 67–71.

Dayhoff, M.O., Schwartz, R.M., and Orcutt, B.C. 1978. A Model of Evolutionary Change in Proteins, 345–358. In Dayhoff, M.O., and Eck, R.V., eds., *Atlas of Protein Sequence and Structure* 5 supp. 3., Natl. Biomed. Res. Found., Washington, DC.

Doolittle, R.F. 1996. *Methods in Enzymology* 266. Academic Press, San Diego, CA.

Galambos, J. 1978. *The Asymptotic Theory of Extreme Order Statistics.* John Wiley & Sons, New York, NY.

Gumbel, E.J. 1958. *Statistics of Extremes.* Columbia University Press, New York, NY.

Henikoff, S., and Henikoff, J.G. 1992. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U.S.A.* 89(22), 10915–10919.

Karlin, S., and Altschul, S.F. 1990. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci. U.S.A.* 87(6), 2264–2268.

Karlin, S., and Dembo, A. 1992. Limit distributions of the maximal segmental score among Markov-

dependent partial sums. *Adv. Appl. Prob.* 24(1), 113–140.

Karlin, S., and Altschul, S.F. 1993. Applications and statistics for multiple high-scoring segments in molecular sequences. *Proc. Natl. Acad. Sci U.S.A.* 90(12), 5873–5877.

Mott, R. 1992. Maximum likelihood estimation of the statistical distribution of Smith-Waterman local sequence similarity scores. *Bull. Math. Biol.* 54(1), 59–75.

Mott, R., and Tribe, R. 1999. Approximate statistics of gapped alignments. *J. Comp. Biol.* 6(1), 91–112.

Mott, R. 2000. Accurate formula for p-values of gapped local sequence and profile alignments. *J. Mol. Biol.* 300(3), 649–659.

Needleman, S.B., and Wunsch, C.D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48(3), 443–453.

Olsen, R., Bundschuh, R., and Hwa, T. 1999. Rapid Assessment of Extremal Statistics for Gapped Local Alignment, 211-222. In Lengauer, T., *et al.*, eds., *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, AAAI Press, Menlo Park, CA.

Pearson, W.R., and Lipman, D.J. 1988. Improved Tools for Biological Sequence Comparison. *Proc. Natl. Acad. Sci. U.S.A.* 85(8), 2444–2448.

Pearson, W.R. 1991. Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. *Genomics* 11(3), 635–650.

Siegmund, D., and Yakir, B. 2000. Approximate p-values for Sequence Alignments. *Ann. Stat.* 28(3), 657–680.

Smith, S.F., and Waterman, M.S. 1981. Comparison of biosequences. *Adv. Appl. Math.* 2, 482–489.

Smith, T.F., Waterman, M.S., and Burks,C. 1985. The statistical distribution of nucleic acid similarities. *Nucleic Acids Research* 13(2), 645–656.

Waterman, M.S., and Vingron, M. 1994a. Rapid and accurate estimates of statistical significance for sequence database searches. *Proc. Natl. Acad. Sci. U.S.A.* 91(11), 4625–4628.

Waterman, M.S., and Vingron, M. 1994b. Sequence Comparison Significance and Poisson Approximation. *Stat. Sci.* 9(3), 367–381.

Waterman, M.S. 1995. *Introduction to Computational Biology.* Chapman & Hall, London, UK.

Yu, Y.K., and Hwa, T. 2001. Statistical Significance of Probabilistic Sequence Alignment and Related Local Hidden Markov Models. *J. Comp. Biol.*, in press.

Zhang, Y. 1995. A Limit Theorem for Matching Random Sequences Allowing Deletions. *Ann. App. Probab.* 5(4), 1236–1240.