Accurate Statistics for Local Sequence Alignment with Position-Dependent Scoring by Rare-Event Sampling

Stefan Wolfsheimer[1], Alexander K. Hartmann[1], and Sven Rahmann[2]

[1] Institute for Theoretical Physics, University Göttingen
Friedrich-Hund-Platz 1, D-37077 Göttingen, Germany

[2] Algorithms and Statistics for Systems Biology group, Genome Informatics
Faculty of Technology, Bielefeld University, D-33594 Bielefeld, Germany
E-mail: `Sven.Rahmann@cebitec.uni-bielefeld.de`

Abstract

Statistics of pairwise local sequence alignment ("how probable is it to observe a certain alignment score on two random sequences of given length?") mainly focuses on the (mathematically convenient) null model where *both* sequences are random i.i.d. objects. In many applications, such as a large-scale database homology search search for a transmembrane protein, this model model is not the most appropriate one: One sequence (the query) remains fixed, while it is scored against many other sequences. Search sensitivity and specificity benefit from position-dependent scoring schemes or use of HMMs. Additionally, one may wish to relax the i.i.d. assumption in the null model. Despite their practical importance, the statistical properties of these settings have not been well investigated yet.

We present an efficient general method that computes (especially the tail) of the exact score distribution to any desired accuracy. It is applicable to many null models and many similarity measures that satisfy a few weak assumptions (e.g., HMMs, normalized alignment, even stochastic context free grammars). Our method uses recent ideas from rare-event simulation, combining Markov chain Monte Carlo simulation with importance sampling and parallel tempering. It significantly extends the class of models where exact p-values can be computed.

While this paper focuses on the methodology, we also present results on alignment statistics for transmembrane protein homology search with position-dependent scoring schemes.

1 Introduction

Sequence-comparison algorithms (e.g., the Smith-Waterman algorithm [20] for pairwise local sequence alignment, or the Viterbi algorithm for sequence-to-HMM alignment [19]) initially return a *raw similarity score* that quantifies the similarity between the input objects, but is hard to interpret because one does not know the absolute scale of the score.

An interpretation becomes possible when we specify a probabilistic null model for the input: Then the similarity score becomes a random variable S whose probabilities $\mathbb{P}(S = s)$ under the null model can be determined (sometimes) analytically or (always) numerically (e.g., by simulation). The *p-value* assigned to an observed score s is defined as $p(s) := \mathbb{P}(S \geq s)$ in the null model, and $-\log p(s)$ is a measure of surprise (and hence a universally normalized score) for s. The key problem is, of course, to find $\mathbb{P}(S)$ for a given comparison method, a given scoring scheme, and a given null model.

In this paper, we introduce an *efficient* and *generally applicable* technique that solves this problem in many different sequence comparison settings, such as for a BLAST-like database search [3] with a fixed query, for position-specific scoring and/or gap-cost schemes (essentially HMMs), or for normalized alignment [5], and in each of those settings, for a variety of null models in addition to the i.i.d. model.

Previous work. Let Σ be a fixed alphabet, e.g., the nucleotides or amino acids.

Most of the existing statistical work for pairwise sequence comparison focuses on null models that assume that both sequences are random and follow a simple model where symbol $c \in \Sigma$ at each position is chosen independently of the other positions, with a given frequency $f_c > 0$, where $\sum_{c\in\Sigma} f_c = 1$ (i.i.d. model; the frequencies often reflect the average composition of the UniProt/SwissProt database [6]). Scores for individual pairs of symbols are given by a constant (i.e., position-independent) symmetric $\Sigma\times\Sigma$ scoring matrix (such as BLOSUM62 [13]) with negative expected score. Gap costs are mostly restricted to linear or affine schemes. We shall refer to this model later as "random query - general-purpose scoring" (RQGS).

For gapless (i.e., infinite gap cost) pairwise local sequence alignment, the raw score distribution can be derived numerically by Markov chain analysis [17] and also analytically in the limit for infinite sequences (Karlin-Altschul or Dembo-Karlin statistics [15]): Properly normalized, it is an extreme value distribution (EVD), also called Gumbel distribution [10]: $\mathbb{P}(S > s) = 1 - \exp[-c \cdot \exp(-\lambda s)]$, where the parameters $\lambda > 0$ and $c > 0$ depend on the score matrix, the symbol frequencies f, and the sequence lengths m and n. In fact, asymptotically we can write $c = Kmn$ for a length-independent constant $K > 0$.

For gapped pairwise local sequence alignment, there exist no universal analytic results, but special cases [9] and empirical evidence also indicate convergence towards the Gumbel form for long sequences; λ and K now additionally depend on the gap cost function [21]. Several recent works have focused on efficient numerical estimation of these parameters [1,7]. The "edge effects" [2] for finite sequences have been dealt with in various ways (e.g., adjusting the lengths of the sequences to "effective lengths", or by adding a Gaussian correction term to the Gumbel distribution [11]).

The (RQGS) model is convenient, because the problem of computing significance values reduces to estimating only two parameters each scoring scheme, which can be pre-computed. However, there are also several problems. For instance, the mathematics do not automatically extend to more complex null models than the i.i.d. model, which is one of the reasons that they are not used in practice. Another striking consequence is the following one: The p-values reported by (the original) BLAST only depend on the raw score and query and subject length, and not on the individual query. This leads to large distortions when the query composition does not match the null model composition. For example, when we run a homology search for the Human transmembrane protein rhodopsin (UniProt accession `P08100`) with BLAST (BLOSUM 62, gap-init 12, gap-extend 1, no composition adjustment, no filtering), we find a possibly remote homolog `Q8NH42` with an E-value of $9 \cdot 10^{-8}$. However, using a recent "composition-based statistics" option [23, 24] leads to a very different E-value of 0.001 for the same protein. This underlines the importance of query-specific or at least composition-based statistics.

The statistics of position-dependent scoring schemes (e.g., as used in PSI-BLAST [4]), or if additionally the gap costs also vary by position (equivalent to a Hidden Markov Model, HMM, framework) are much less well explored. The central question here is, "given a fixed query Q and a position-specific scoring scheme, what is the score distribution when random null-model sequences of given length are scored against `Q`?". We refer to this model as "fixed query - position-dependent scoring" (FQPS). Here EVDs are still used heuristically, and one attempts to fit the parameters by straightforward sampling from the score distribution by generating random sequences (implemented in, e.g., the `hmmcalibrate` program from the HMMER package [8]). Such an approach may not give the tail of the distribution correctly, although this is most important for the estimation of statistical significance.

Our motivation for simulation-based method that initially makes no parametric assumption comes from a paper [18] on increasing the sensitivity of detecting homologs of a given transmembrane (TM) protein in a database search: Using a bipartite scoring scheme with a (non-symmetric) transmembrane helix specific scoring matrix (such as SLIM [18]) for the TM helices and a general-purpose scoring matrix (such as BLOSUM [13]) for the remaining regions of the query protein — see Figure 1 — results in higher search sensitivity and/or specificity. However, a statistical theory

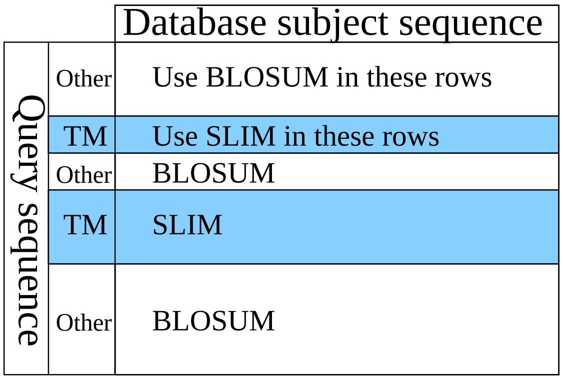

Figure 1: Bipartite scoring scheme for detection of homologous transmembrane proteins from Ref. [18]. The figure represents the Smith-Waterman alignment matrix and indicates which scoring matrix is used for which query positions (rows): In transmembrane helices, a transmembrane-specific scoring matrix is used. For p-value computations, the query is assumed fixed and the subject is assumed a random i.i.d. sequence drawn from the amino acid frequencies of the database.

or efficient computational method in such an (FQPS) framework has been missing.

Our contributions and paper outline. We present a general framework (Algorithm 1) for efficient estimation of the tail of raw score distributions in sequence comparison problems. We only make the following assumptions: (1) We can sample sequences x according to the null model and compute the null model probability of any given x. (2) We have an efficient algorithm $\mathcal{A}$ that computes the score $S(x)$ (where x could be a pair of random sequences in (RQGS), or one fixed and one random sequence in (FQPS)). (3) The scores are rational numbers with a common denominator (without loss of generality, they can be assumed to be integers).

Our framework is readily applicable to the (RQGS) and (FQPS) models, but also to more exotic settings, such as normalized alignment [5], where the score is not additive, but normalized by the alignment length, for which no statistical framework exists so far.

The paper is now organized as follows. Section 2 presents the mathematical background on importance sampling and Markov chain Monte Carlo methods. Section 3 describes our method. For ease of exposition, we first state it in terms of the local alignment score for a fixed query and position-dependent score matrix (as illustrated in Figure 1), but then provide a general high-level description to underline its versatility. Section 4 shows computational results on transmembrane protein similarity statistics in (RQGS) and (FQPS). A discussion in Section 5 concludes the paper.

2 Background

Importance sampling. Importance sampling is a general technique to reduce the variance in the estimation of quantities that can be written as an expectation $\mathbb{E}[h(X)]$, where X is a random object and h is a real function. We assume that we can draw n random samples $X_1, \dots, X_n$ from the null model. The expectation is then approximated by the empirical mean $\mathbb{E}[h(X)] \approx 1/n \cdot \sum_{i=1}^{n} h(X_i)$.

In our setting, to estimate the score distribution (and then p-values), we consider a sequence space $\mathcal{X}$, from which we generate n random sequences $X_1, \dots, X_n$ and align them to the query by a given algorithm and note the resulting similarity scores $S(X_i)$. We consider the family of functions $h_s : \mathcal{X} \to \mathbb{R}$ for all $s > 0$, defined by $h_s(x) := 1$ if $S(x) = s$, and $h_s(x) := 0$ if $S(x) \neq s$; so $\mathbb{E}[h_s(X)] = \mathbb{P}(S(X) = s) \approx |\{i : S(X_i) = s\}|/n$.

If the probability to be estimated is small, say 10^{-9}, we need about 10^{12} samples to estimate it with reasonable precision. For very rare events, this "naive" sampling quickly becomes infeasible.

Importance sampling generates the "interesting" events more often by sampling from a different distribution and correcting for this bias afterwards, which results in a more accurate estimate with fewer samples. Let p be the probability mass function (pmf) of X, and let q be another pmf satisfying $q(x) > 0$ whenever $p(x) > 0$. Then

$$\mathbb{E}[h(X)] = \sum_{x} h(x) \cdot p(x) = \sum_{x'} h(x') \cdot \frac{p(x')}{q(x')} \cdot q(x') = \mathbb{E}\left[h(X')\frac{p(X')}{q(X')}\right] \approx \frac{1}{n}\sum_{i=1}^{n} h(X'_i) \cdot \frac{p(X'_i)}{q(X'_i)}, \quad (1)$$

where each X'_i is sampled from pmf q. To successfully apply importance sampling, q has to fulfill three properties: First, it needs to put high probability on the region of interest (where $h \cdot p$ is high); second, we need to be able to sample according to q; third, we need to be able to compute the correcting weight $p(x)/q(x)$. Since direct sampling from q often proves difficult, we shall use a general sampling method, which we describe next.

Metropolis-Hastings sampling. If we need to generate samples from a discrete distribution q but have not easy direct method to do so, the Metropolis-Hastings method [12] provides a solution by constructing an ergodic Markov chain with stationary distribution q in the following way.

Let us call the elements of the sample space $\mathcal{X}$ *configurations*. Each configuration x has a set $N(x)$ of potential neighbors $x' \in N(x)$ that are proposed with positive probability $P_{x,x'}$ as the next configuration. The proposal is accepted with probability $\min\left\{1, \frac{q(x')\cdot P_{x',x}}{q(x)\cdot P_{x,x'}}\right\}$, in which case x' becomes the new configuration x; otherwise x' is discarded and x is unchanged.

For an appropriate choice of neighborhoods $N(x)$ and of $P_{x,x'}$, the so-constructed Markov chain is indeed ergodic, and the distribution of the configuration converges exponentially fast towards q, irrespective of the start configuration. We say that the chain has reached *equilibrium* when convergence has occurred up to numerically negligible error. Thus, if the configuration x is sampled after equilibration, it will behave like as a sample from q. In practice, the exact speed of equilibration is unknown and convergence diagnostics are applied (see below). Several (almost) independent samples are obtained by running the chain further and taking a sample every k-th step for sufficiently large k to allow time for "forgetting" the state of the last sampled configuration.

Parallel tempering. Instead of importance sampling from a single distribution q, we sample from several distributions q_T for different values of a parameter T (which in physical systems is often called "temperature"). It is beneficial to perform this sampling not sequentially, but to run the corresponding Markov chains in parallel. This has two advantages. First, we can re-use some random numbers. Second, the configurations of different chains can communicate with each other: After some steps, we propose to swap the configurations of two chains with neighboring T-values and accept the proposal with an appropriately chosen probability. This step preserves the stationary distribution and ergodicity of the Markov chains and has proven to accelerate equilibration and mixing [14, 16]. This technique is known as parallel tempering or replica exchange.

3 Methodology

We first focus on the score distribution for a fixed query with position-dependent scores (including gap costs) against random i.i.d. sequences of given length (FQPS). This setting applies to the transmembrane protein homology search, as well as to alignments of sequences against a profile HMM, and may thus be of practical importance as a replacement for `hmmcalibrate`. Further below we present a more general and abstract description of the method.

Statistics for pairwise local alignment with position-dependent scores. We are given a query sequence y and a position-specific scoring scheme or a profile HMM y and an algorithm $\mathcal{A}$ to compute a similarity score $S(x)$ of any sequence against y. We fix a length m for x, and for simplicity, we assume that $x = (x_1, \ldots, x_m) \in \Sigma^m$ is drawn from an i.i.d. model (but this assumption is not necessary!) given by character frequencies $(f_c)_{c\in\Sigma}$, such that $p(x) = \prod_{i=1}^m f_{x_i}$.

Our method importance-samples different regions from the tail of the score distribution in parallel; thus we need to increase the probability of high-scoring random sequences X. We do so by introducing a parameterized family of tilted random sequence models q_T ($T \geq 0$), defined by

$$q_T(x) := p(x) \cdot \exp(S(x)/T)/Z_T, \tag{2}$$

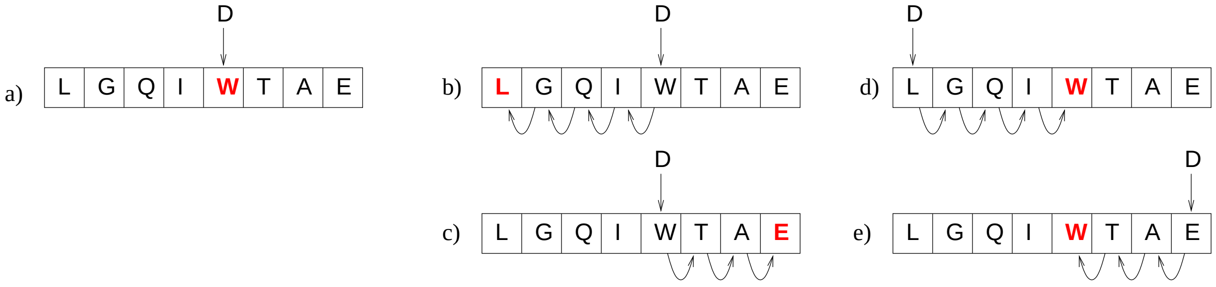

Figure 2: Monte Carlo moves used in the simulation: (a) substitution, (b) insertion with left shift, (c) insertion with right shift,(d) deletion with right shift and (e) deletion with left shift.

where $Z_T := \sum_x p(x) \cdot \exp(S(x)/T)$ is the normalization constant ensuring that q_T is a probability distribution. Note that Z_T cannot be explicitly computed since the sequence space is too large. Also note that $q_T \to p$ as $T \to \infty$ and $Z_\infty = 1$. According to Eq. (1), the weight for samples from q_T is $p(x)/q_T(x) = Z_T \cdot \exp(-S(x)/T)$ that involves Z_T, a problem which we deal with later. The choice of the factor $\exp(S(x)/T)$ is motivated by the observation that if $T = 1/\lambda$ for the λ in the Gumbel distribution, then $\mathbb{P}(S(X) = s) \cdot e^{s/T}$ cancels out the dominating $e^{-s/T}$ factor leading to an "almost flat" histogram of sampled scores. Note that the choice is arbitrary, though: Any function that enforces the system to reach the high score region by simulation will also work.

We need to define the neighborhoods $N(x)$ and the proposal probabilities $P_{x,x'}$. A simple way is to let $N(x) := \{x' \mid x \text{ and } x' \text{ differ at most at one position}\}$, where we randomly select a position i from $\{1, \ldots, m\}$ and replace x_i by another letter c, taken randomly according to (f_c); see Fig. 2a). A wider neighborhood definition, based on random character insertions and deletions illustrated in Fig. 2 a)–e) considerably accelerates equilibration. In order to keep the sequence length fixed, one character will be deleted / inserted at one of the sequence ends when a random character is inserted / deleted in the middle. The acceptance probability of such a move is then

$$\min\left\{1, \frac{q_T(x') \cdot P_{x',x}}{q_T(x) \cdot P_{x,x'}}\right\} = \min\left\{1, \frac{e^{S(x')/T} \cdot p(x') \cdot P_{x',x}}{e^{S(x)/T} \cdot p(x) \cdot P_{x,x'}}\right\} = \min\{1, \exp[(S(x') - S(x))/T]\},$$

since $p(x') \cdot P_{x',x} = p(x) \cdot P_{x,x'}$ by construction.

In order to sample properly from q_T, the chain has to run until it reaches equilibrium and the effects of the initial state have worn off. To detect the equilibration time, we start the chain from two extreme starting configurations: from a low-scoring x^0 that we chose randomly according to the null model p, and from a high-scoring x^* that is identical to the query y.[1] For our purposes, equilibrium is reached when the score of both chains has moved into the same range. To detect this, a moving score average and standard deviation is computed for each chain and tested for equality (within error bars). For small T, equilibration takes prohibitively long because the system gets stuck in local high score regions [11]. As described above, parallel tempering remedies some of these problems.

As consecutive configurations are highly correlated, we cannot sample the score in every step, but wait until the configuration has become sufficiently independent from the last sample; this time is called the mixing time of the chain. Parallel tempering also improves mixing. In our setting, we found that typical values for the equilibration time are about 10^5 steps, attempting a replica flip every 10th step. Samples are taken at every 100th step after equilibration.

Sampling continues until we obtain a pre-defined number n of samples. For each temperature T, we obtain score counts $C_T[s] := |\{\text{samples with score } s\}|$ and an empirical distribution (histogram)

[1] Since in general x and y do not have the same length, it is not possible to take exactly y as a starting configuration. Therefore only a central subsequence of the longer sequence is chosen to be equal to the shorter one.

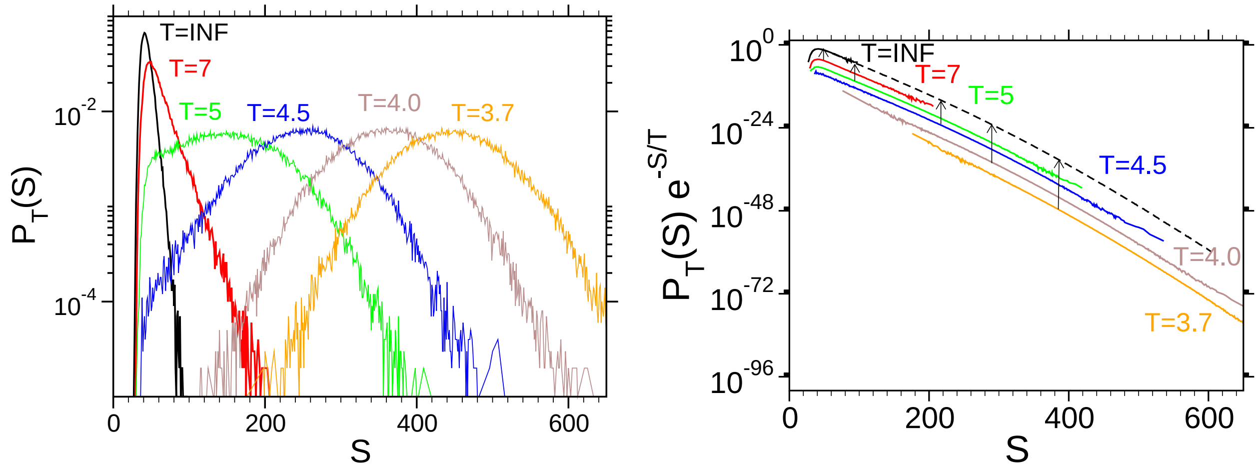

Figure 3: Illustration of the reweighting procedure for the score distribution of the transmembrane protein `YXX5_CAEEL` (putative neuropeptide Y receptor; `Q18179`) of length 455. The bipartite scoring scheme from Figure 1 was applied to random i.i.d. sequences of typical UniProt composition, of the same length 455. Left: Raw empirical distributions $P_T(S)$ obtained by sampling the score for different temperatures T, based on 120 000 samples for each temperature. Right: Reweighted distributions. The slopes in overlapping regions match, and the normalization constants Z_T still need to be determined by shifting the curves upward for $T < \infty$, matching them in overlapping regions, proceeding from left to right.

$P_T(s) := C_T[s]/n$. In the score interval where the sample count in C_T is high, the score distribution $\mathbb{P}(S = s)$ is well estimated by $P_T(s) \cdot e^{-S/T} \cdot Z_T$, which still contains the unknown Z_T. This is not a problem, however, if these intervals overlap for neighboring values of T, since we know that $Z_\infty = 1$: Assume that we sampled at N temperatures $T \in \{\infty = T_0 > T_1 > \cdots > T_N\}$ and have already determined $Z_{T_0} = 1, Z_{T_1}, \ldots, Z_{T_k}$ (initially $k = 0$). Then we can determine $Z_{T_{k+1}}$ by fitting $P_{T_{k+1}}(s)$ to $P_{T_k}(s)$ in the overlapping region [11, 22]. An example is shown in Figure 3.

General framework. The above methodology is in fact much more general. We may replace the i.i.d. assumption on p by something more complex; care needs to be taken that the acceptance probability can still be efficiently computed. The tilting factor $\exp(S(x)/T)$ can be replaced by a problem-specific function $f_T(S(x))$ that is increasing in S and decreasing in T. The sequence space $\mathcal{X}$ and scoring algorithm $\mathcal{A}$ can be adjusted to fit almost any similarity scoring algorithm from sequence analysis. For example, our framework is probably the first one that can efficiently deal with the statistics of *normalized sequence alignment* [5] or stochastic context-free grammars. The general framework is shown in Algorithm 1.

Of course, it also fits classical local alignment statistics (RQGS), where two random sequences $x = (x^1, x^2)$ of lengths m_1, m_2 are considered. Here $\mathcal{X} = \Sigma^{m_1+m_2}$ and $p(x) = \prod_{i=1}^{m_1} f_{x_i^1} \cdot \prod_{i=1}^{m_2} f_{x_i^2}$. Steps in the Markov chain might involve changing one character in one sequence at a time. In this context, the score distribution is analyzed in depth by [11, 22].

4 Results

To our knowledge, we present the first highly accurate score statistics for alignments with position-specific scoring schemes.

Shape of the score distribution. If the score distribution follows a Gumbel form $\mathbb{P}(S > s) = 1 - \exp[-c \cdot \exp(-\lambda s)]$, then, in the far right tail, essentially $\mathbb{P}(S > s) = c \cdot \exp(-\lambda s)$

Algorithm 1 General sequence similarity score sampling framework. It is assumed that the following are available: sequence space $\mathcal{X}$, null distribution $p(x)$ with efficient algorithms to sample from p and to compute $p(x)$ for any $x \in \mathcal{X}$, neighborhoods $N(x)$ with proposal distributions $P_{x,x'}$ for each x and a way to draw samples from $P_{x,\cdot}$, scoring algorithm $\mathcal{A} : \mathcal{X} \to \mathbb{Z}$ with $x \mapsto S(x)$, tilting factors $f_T(s)$. For simplicity, this overview does not show parallel tempering.

ComputeScoreHistogram(T, n)
Input: Temperature T, number of samples n
Output: Score counts $C[s] :=$ number of times that score s is sampled, for all s
$C[s'] \leftarrow 0$ for all possible score values s'
$\triangleright$ *Initialize and equilibrate the Markov chain for sampling from* $q_T := p \cdot f_T$.
Pick any $x \in \mathcal{X}$; compute its null probability $p := p(x)$; compute $s := S(x)$ using $\mathcal{A}$.
repeat
    $(x, p, s) \leftarrow$ MetropolisHastingsUpdate(x, p, s)
until equilibration has occurred
$\triangleright$ *Obtain* n *samples from* q_T *and their score counts / histogram*
for $i = 1..n$ **do**
    $C[s] \leftarrow C[s] + 1$
    repeat
        $(x, p, s) \leftarrow$ MetropolisHastingsUpdate(x, p, s)
    until mixing has occurred
end for
return counts C $\triangleright$ *Histogram* $P_T(s) := C[s]/n$

MetropolisHastingsUpdate(x, p, s)
Input: Sequence x, its null probability $p = p(x)$ and score $s = S(x)$.
Output: Possibly new values for x, p, s.
Draw $x' \in N(x)$ with probability $P_{x,x'}$; compute $p' := p(x')$; compute $s' := S(x')$ using $\mathcal{A}$.
Compute $\alpha := \frac{f_T(s') \cdot p' \cdot P_{x',x}}{f_T(s) \cdot p \cdot P_{x,x'}}$. $\triangleright$ *Design such that* $p(x') \cdot P_{x',x} = p(x) \cdot P_{x,x'}$
With probability $\min\{1, \alpha\}$: Let $(x, p, s) \leftarrow (x', p', s')$
return (x, p, s)

since $1 - \exp[-\varepsilon] = \varepsilon$ (numerically) for very small $\varepsilon > 0$. Hence $\mathbb{P}(S = s) = c' \cdot \exp(-\lambda s)$ (with $c' = c \cdot (e^{\lambda} - 1)$), and $\log \mathbb{P}(S = s)$ should be an affine function $s \mapsto -\lambda \cdot (s - s_0)$ with $s_0 := -\log(c')/\lambda$.

As pointed out by Altschul and Gish [2], edge effects occur for finite sequences: An alignment may extend to the end of either sequence and the score will be distorted towards lower values and high scores become less probable. In the limit of infinite sequences this effect vanishes and the tail of the Gumbel distribution can be understood as an upper bound for finite sequences. Indeed, we clearly see that the curves in Figure 3 (right) are not straight lines in the right tail, but have negative curvature.

A better fit with the empirical distribution is obtained by determining parameters s_0, $\lambda > 0$, $\lambda_2 > 0$ in a modified Gumbel distribution with Gaussian correction,

$$\log \mathbb{P}(S = s) = \log(\lambda) - \lambda\,(s - s_0) - \lambda_2\,(s - s_0)^2\,, \tag{3}$$

where s_0 can be interpreted as the center of the distribution. The parameter λ_2 is generally small (and thus shows its effect only in the far tail). It vanishes for sequences of equal length as the length tends to infinity. Previously, such a Gaussian correction has been proposed for for (RQGS) statistics and computed for one parameter set of BLOSUM62 with affine gap costs [11][2].

[2] An systematic investigation of a larger number of parameters has been submitted [22].

ID	AC	Description	Organism	Length
OPSD_HUMAN	P08100	Rhodopsin	H. sapiens	348
AGTR2_HUMAN	P50052	type-2 angiotension II receptor	H. sapiens	363
YXX5_CAEEL	Q18179	putative neuropeptide Y receptor	C. elegans	455
ADA1A_HUMAN	P35348	Alpha-1A adrenergic receptor	H. sapiens	466

Table 1: A selection of transmembrane proteins. ID: UniProt identifier; AC: accession number.

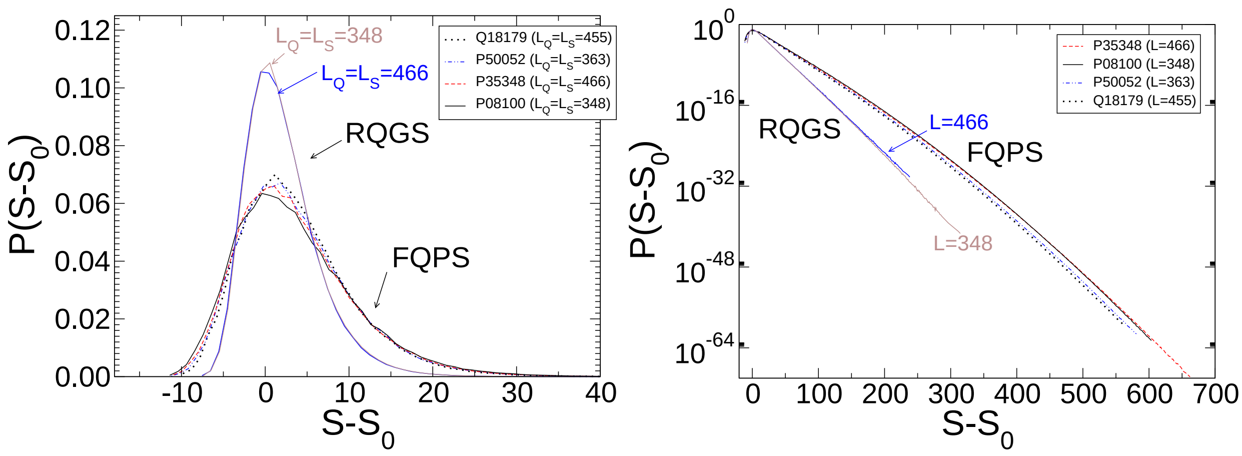

Figure 4: Score distributions for (RQGS = classical) and (FQPS) models where the subject length equals the query length. In order to compare the shape, the distributions have been shifted by the center s_0. Left: Linear view; all distributions from the (RQGS) agree outside the tails (only two lengths are shown). The shape of the (FQPS) distributions is more variable. Right: Logarithmic view; significant differences between the two models appear in the tail of the distribution. High scores are more probable for the (FQPS) alignment. Furthermore the curvature, i.e. the deviation from the Gumbel form, is much larger for (FQPS) than for the classical model.

Here we observe that the curvature is more pronounced in the (FQPS) model: We discuss four different transmembrane proteins as queries (see Table 1) in both the (RQGS) or classical model and the (FQPS) model. We use Smith-Waterman alignment with the BLOSUM62 matrix (BLOSUM62 and SLIM using the bipartite scheme from Figure 1 in (FQPS)), affine gap costs with a gap-open penalty of 12 and a gap-extension penalty of 1, and an i.i.d. random model with UniProt frequencies. The subject length is set to the query length.

Significant differences of shapes already show up in the high probability region, which is accessible by simple sampling (Figure 4 left). All (RQGS) distributions match almost perfectly (only two lengths are shown), whereas the shape of the (FQPS) distributions (2) varies slightly with the sequence type. This shows that position-specific scoring in connection with a fixed query sequence is better able to discriminate between different sequences than the standard approach of having two random sequences in connection with position-independent scoring, as already found in [18].

More pronounced differences are seen in the behavior of the tail (Figure 4 right), which is only accessible via our importance sampling approach. The difference between the probabilities spans several orders of magnitude; hence a wrong choice of the model would falsify the estimation of significance drastically. Most importantly, the pmf obtained using the position-specific scoring is considerably curved. Thus, using EVDs from fits to data of the high-probability region is even more questionable here than in the (RQGS) model, where the pmf is almost a straight line.

Parameters of the modified Gumbel distribution. To investigate the impact of the the query and subject lengths $L_\mathtt{Q}$ and $L_\mathtt{S}$ on the parameters of the modified Gumbel distribution, we vary $L_\mathtt{S}$ and consider the parameters λ and λ_2 as functions of the ratio $L_\mathtt{S}/L_\mathtt{S}$ (see Figure 5). The

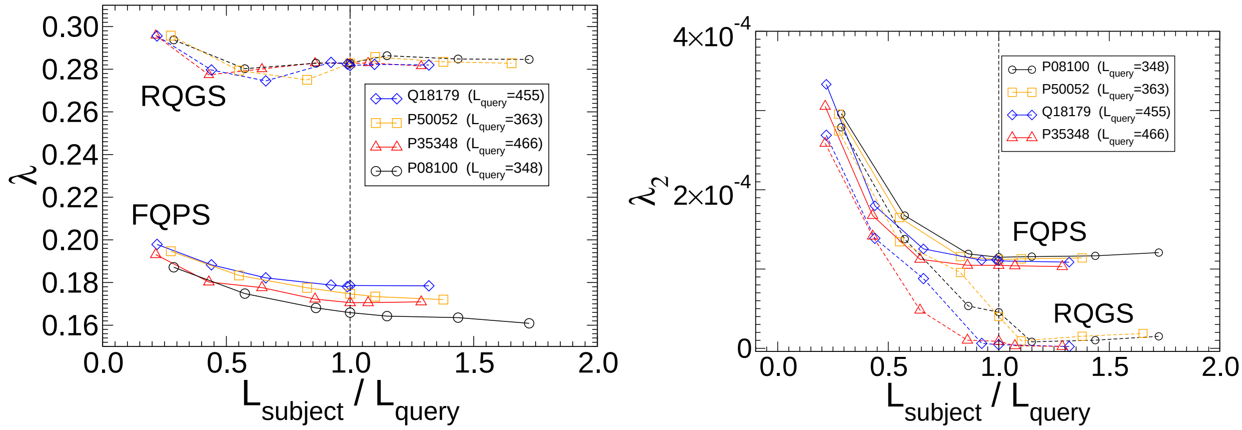

Figure 5: Dependence of the modified Gumbel parameters on the subject/query length ratio $L_{\mathtt{S}}/L_{\mathtt{Q}}$. The vertical line corresponds to Figure 4, where $L_{\mathtt{S}} = L_{\mathtt{Q}}$. Left: λ describes the bulk of the distribution (see Figure 4 left). For $L_{\mathtt{S}} > L_{\mathtt{Q}}$, λ varies only slightly in the subject length. Right: The parameter λ_2 characterizes the curvature of the pmf in the tail (see 4 right). Large differences between (RQGS) and (FQPS) show up in the case where $L_{\mathtt{S}} > L_{\mathtt{Q}}$. λ_2 becomes subject-length independent for $L_{\mathtt{S}} > L_{\mathtt{S}}$.

large gap between the values of λ for the two different models reflects the qualitative difference of the shape in the high probability regime. We see that in model (RQGS), λ is virtually independent of query and sequence length. However, in model (FQPS), λ varies with each individual query, as has to be expected. For λ_2 one has to distinguish between $L_{\mathtt{S}} < L_{\mathtt{Q}}$ and $L_{\mathtt{S}} > L_{\mathtt{Q}}$. In the first case, λ_2 decreases, which is not surprising, since the Gaussian correction describes a finite-size effect and should vanish for increasing sequence lengths. Once the target length exceeds the query length, the search space is still growing, but the finite length of the query enforces target size independent edge effects. Appendix A provides numerical values for reference.

5 Concluding Discussion

We have presented a simple universal method to accurately sample the far right tails of the score distribution of many sequence comparison algorithms. It appears to be the first method that is applicable to all of classical local alignment statistics, query-specific and position-dependent score statistics, HMM calibration, statistics of normalized alignments, and many more. We need no initial parametric assumptions, but can a posteriori to fit the observed distribution to an appropriate parametric form. Here we observed that for the (FQPS) model, the Gumbel distribution should be replaced by a more negatively curved one.

The method has a disadvantage: Because of the high number of samples required for non-parametric estimation of the distribution, it can presently not be used in on-line database search web services, such as a BLAST server. For example, to generate the 720 000 samples for Figure 3 took approximately 100 hours on an Intel Pentium 4 with 1.8 GHz. This is not as bad as it seems, though: Both the implementation and the design of the Markov chain have much room for improvement (e.g., we can choose different neighborhoods $N(x)$ and optimize the set of temperatures for parallel tempering). We can probably reduce the required time for sufficient accuracy to a few hours.

While this still prohibits interactive use, we see a lot of potential for our method to provide an improved version of the `hmmcalibrate` tool [8] and to explore the statistics of normalized sequence alignment [5].

Acknowledgments. SW and AKH are supported by the German *VolkswagenStiftung* (program “Nachwuchsgruppen an Universitäten”) and by the European Community DYGLAGEMEM pro-

gram.

References

[1] S. F. Altschul, R. Bundschuh, R. Olsen, and T. Hwa. The estimation of statistical parameters for local alignment score distributions. *Nucleic Acids Res*, 29(2):351–361, 2001.

[2] S. F. Altschul and W. Gish. Local alignment statistics. *Meth. Enzym.*, 266:460, 1996.

[3] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3):403–410, 1990.

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

[5] A. N. Arslan, O. Egecioglu, and P. A. Pevzner. A new approach to sequence comparison: normalized sequence alignment. *Bioinformatics*, 17(4):327–337, 2001.

[6] A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi, and L.-S. L. Yeh. The Universal Protein Resource (UniProt). *Nucleic Acids Res*, 33(Database issue):D154–D159, 2005.

[7] N. Chia and R. Bundschuh. A practical approach to significance assessment in alignment with gaps. *J Comput Biol*, 13(2):429–441, 2006.

[8] S. Eddy. HMMER User's guide, version 2.3.2, 2003. `ftp://selab.janelia.org/pub/software/hmmer/CURRENT/Userguide.pdf`.

[9] S. Grossmann and B. Yakir. Large deviations for global maxima of independent superadditive processes with negative drift and an application to optimal sequence alignments. *Bernoulli*, 10(5):829–845, 2004.

[10] E. J. Gumbel. *Statistics of Extremes*. Columbia University Press, New York, 1958.

[11] A. K. Hartmann. Sampling rare events: statistics of local sequence alignments. *Phys. Rev. E*, 65:056102, 2002.

[12] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[13] S. Heinkoff and J. G. Heinkoff. Amino acid substitution matrices from protein blocks. *Proc.Natl.Acad.Sci.U.S.A.*, 89:10915–10919, 1992.

[14] K. Hukushima and K. Nemoto. Exchange Monte Carlo method and application to spin glass simulations. *J. Phys.Soc.Jpn.*, 65:1604, 1996.

[15] S. Karlin and S. F. Altschul. 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, 1990.

[16] E. Marinari and G. Parisi. Simulated tempering: A new Monte Carlo scheme. *Europhys.Lett.*, 19:451, 1992.

[17] S. Mercier and J. J. Daudin. Exact distribution for the local score of one i.i.d. random sequence. *J Comput Biol*, 8(4):373–380, 2001.

[18] T. Müller, S. Rahmann, and M. Rehmsmeier. Non-symmetric score matrices and the detection of homologous transmembrane proteins. *Bioinformatics*, 17:S182, 2001.

[19] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[20] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–197, 1981.

[21] M. S. Waterman and M. Vingron. Rapid and accurate estimates of statistical significance for sequence data base searches. *Proc Natl Acad Sci U S A*, 91(11):4625–4628, 1994.

[22] S. Wolfsheimer, B. Burghardt, and A. K. Hartmann. Local sequence alignments statistics: Deviations from gumbel statistics in the rare-event tail. *To be published*, 2006.

[23] Y.-K. Yu and S. F. Altschul. The construction of amino acid substitution matrices for the comparison of proteins with non-standard compositions. *Bioinformatics*, 21(7):902–911, 2005.

[24] Y.-K. Yu, J. C. Wootton, and S. F. Altschul. The compositional adjustment of amino acid substitution matrices. *Proc Natl Acad Sci U S A*, 100(26):15688–15693, 2003.

A Appendix: Modified Gumbel Parameters

Tables 2 and 3 show numerical values for the parameters λ and λ_2 of the modified Gumbel distribution Eq. (3). These are visualized in Figure 5 in the body of the paper.

Query	$L_\mathtt{S}$	λ	$10^4\lambda_2$	S_0
P08100	100	0.187(2)	2.96(0)	33.2(0)
$L_\mathtt{Q} = 348$	200	0.174(8)	1.67(4)	37.2(8)
	300	0.168(1)	1.19(0)	39.5(7)
	348	0.165(9)	1.14(9)	40.4(0)
	400	0.164(3)	1.15(6)	41.1(3)
	500	0.163(5)	1.16(7)	42.7(0)
	600	0.160(9)	1.12(1)	43.5(7)
P50052	100	0.194(7)	2.95(5)	32.4(0)
$L_\mathtt{Q} = 363$	200	0.183(3)	1.65(0)	36.1(0)
	300	0.177(5)	1.15(8)	38.0(7)
	363	0.174(7)	1.11(9)	39.2(0)
	400	0.173(4)	1.12(8)	39.9(2)
	500	0.172(0)	1.14(0)	41.1(8)
Q18179	100	0.197(9)	3.33(0)	31.9(5)
$L_\mathtt{Q} = 455$	200	0.188(3)	1.79(8)	35.6(1)
	300	0.182(2)	1.25(4)	37.1(9)
	420	0.178(8)	1.11(1)	39.4(4)
	450	0.178(1)	1.11(6)	39.8(3)
	455	0.178(6)	1.10(2)	39.8(2)
	600	0.178(4)	1.08(7)	41.6(6)
P35348	100	0.193(1)	3.05(3)	33.4(0)
$L_\mathtt{Q} = 466$	200	0.180(3)	1.67(6)	36.9(3)
	300	0.177(6)	1.12(3)	39.4(6)
	400	0.172(3)	1.04(7)	40.9(5)
	466	0.170(6)	1.04(6)	41.9(7)
	500	0.170(6)	1.04(1)	42.2(7)
	600	0.170(9)	1.03(1)	43.5(6)

Table 2: Fitted parameters for four fixed transmembrane queries and position-specific scoring.

$L_{\mathtt{Q}}$	$L_{\mathtt{S}}$	λ	$10^4\lambda_2$	S_0
348	100	0.293(8)	2.7(9)	25.8(6)
	200	0.280(3)	1.3(8)	28.2(6)
	300	0.282(8)	0.5(3)	29.9(0)
	348	0.282(9)	0.4(6)	30.4(5)
	400	0.286(3)	0.0(8)	31.0(7)
	500	0.284(8)	0.1(0)	31.8(6)
	600	0.284(6)	0.1(5)	32.5(6)
363	100	0.295(6)	2.7(4)	26.0(1)
	200	0.279(4)	1.3(4)	28.3(6)
	300	0.275(0)	0.9(6)	29.7(9)
	363	0.282(8)	0.4(0)	30.8(0)
	400	0.285(7)	0.9(4)	31.1(6)
	500	0.283(5)	0.1(5)	32.0(2)
	600	0.282(8)	0.1(8)	32.7(2)
455	100	0.295(6)	2.6(9)	26.8(0)
	200	0.279(6)	1.3(9)	29.2(3)
	300	0.274(5)	0.8(8)	30.6(0)
	420	0.283(2)	0.0(6)	32.2(4)
	450	0.282(4)		32.4(4)
	455	0.281(7)	0.0(4)	32.4(9)
	500	0.282(3)		32.8(2)
	600	0.282(0)	0.0(4)	33.5(9)
466	100	0.295(9)	2.5(9)	26.8(9)
	200	0.277(4)	1.4(2)	29.3(0)
	300	0.280(1)	0.4(8)	30.9(3)
	400	0.282(9)	0.1(0)	32.1(4)
	466	0.282(7)	0.0(9)	32.7(1)
	500	0.283(0)		32.9(7)
	600	0.281(5)		33.5(7)

Table 3: Fitted parameters for different query and subject lengths in the classical statistical setting.