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

Stefan Wolfsheimer[1] , Inke Herms[2] , Sven Rahmann[3] , Alexander K Hartmann[1]

[1]Institut für Physik, Universität Oldenburg, D-26111 Oldenburg, Germany
[2] Technische Fakultät, Universität Bielefeld, 33501 Bielefeld, Germany
[3] Department of Computer Science, TU Dortmund, Germany

Email: Stefan Wolfsheimer - wolfsh@theorie.physik.uni-oldenburg.de ; Inke Herms - ihildebr@cebitec.uni-bielefeld.de ; Sven Rahmann - Sven.Rahmann@tu-dortmund.de ; Alexander K Hartmann - alexander.hartmann@uni-oldenburg.de;

*Corresponding author

Abstract

Background: Statistics of pairwise local sequence alignment involves the question "how probable is a certain alignment score if we compare two random sequences of given length?". So far, research has focused 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 for transmembrane proteins, this 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.

Results: We use here an efficient general method that computes the exact score distribution to any desired accuracy. In this way, we have access to the "tail" of the distribution, describing in particular the region of intermediate score values, which are relevant for practical applications. The method is applicable to many null models and many similarity measures that satisfy a few weak assumptions. It can be applied, e.g., to HMMs, normalized alignment, or even stochastic context-free grammars. Our method uses recent ideas from rare-event simulation, combining Markov chain Monte Carlo simulation with importance sampling and generalized ensembles. It significantly extends the class of models where exact p-values can be computed.

Conclusions: This paper explains the methodology and presents results on alignment statistics for transmembrane protein homology search with position-dependent scoring schemes. We compare the results for random i.i.d. sequences with those for fixed queries. The third approach models queries by a transmembrane HMM. This is a generic random string model adapted to transmembrane proteins through the classification of amino acids to cytoplasmic, membrane, and non-cytoplasmic regions.

Introduction

The most popular sequence-comparison algorithms are the Smith-Waterman algorithm [1] for pairwise local sequence alignment and the Viterbi algorithm for sequence-to-HMM alignment [2]. They return a *raw similarity score* that quantifies the similarity between the input objects. Unfortunately, this raw score 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 $\text{Prob}(S = s)$ under the null model can be determined. Sometimes this can be done analytically, but usually one has to apply numerical simulation. The *p-value* assigned to an observed score s is defined as $pval(s) := \text{Prob}(S \geq s)$ in the null model, and $-\log pval(s)$ is a measure of surprise (and hence a universally normalized score) for s. The key problem is, of course, to find $\text{Prob}(S = s)$ for a given comparison method, a given scoring scheme, and a given null model.

In this paper, we explain and extend 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 [4]. In each of those settings a variety of null models in addition to the i.i.d. model is possible. Before we state our main contributions, we review existing results and methods and illustrate some of their deficiencies, motivating the need for new methods.

Previous work.

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

Most of the existing statistical work for pairwise sequence comparison focuses on null models where both sequences are random and at each position a symbol $\sigma \in \Sigma$ is chosen independently of the other positions ("i.i.d. model"), with a given frequency $f_\sigma > 0$ ($\sum_{\sigma\in\Sigma} f_\sigma = 1$). f often reflects the average composition of proteins in the UniProt/SwissProt database [5]. Scores for individual pairs of symbols are given by a constant (position-independent) symmetric $\Sigma \times \Sigma$ scoring matrix with negative expected score, such as BLOSUM62 [6]. 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 pairwise local sequence alignment, the raw score distribution can be derived numerically by Markov chain analysis [7] and also asymptotically for infinite sequences (Karlin-Altschul or Dembo-Karlin statistics [8]): It is an extreme value distribution (EVD), also called Gumbel distribution [9]: $\mathrm{Prob}(S > s) = 1 - \exp[-c \cdot \exp(-\lambda s)]$, where the parameters $\lambda > 0$ and $c > 0$ depends on the score matrix, on the symbol frequencies f, and on the query and subject sequence lengths $L_{\mathtt{Q}}$ and $L_{\mathtt{S}}$. Asymptotically we have $c = KL_{\mathtt{Q}}L_{\mathtt{S}}$ for a length-independent $K > 0$.

For gapped pairwise local sequence alignment, there exist no universal analytic results, but special cases [10] and empirical evidence also indicate convergence towards the Gumbel form for long sequences; λ and K now additionally depend on the gap-cost function [11]. Several recent works have focused on efficient numerical estimation of these parameters [12]. The "edge effects" [13] for finite sequences are treated in various ways, e.g. by adjusting the lengths of the sequences to "effective lengths" but still assuming a Gumbel form of the distribution. Nevertheless, for moderate sequence lengths, which are biologically most relevant, the true distribution differs strongly from a Gumbel form [14, 15], which can be dealt with by including a correction term to the Gumbel form.

The (RQGS) model is convenient, because the problem of computing significance values reduces to the estimation of only two parameters, which can be precomputed for each scoring scheme. 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 [1] of $9 \cdot 10^{-8}$. However, using a recent "composition-based adjustment" option [16, 17] 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, particularly for intermediate p-values.

The statistics of position-dependent scoring and/or gap-cost schemes, as used in PSI-BLAST [18] or in hidden Markov model (HMM) frameworks, are much less well explored. The central question here is, "given a 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). As compromise between the general (RQGS) and the very specific (FQPS) models, one may release the i.i.d. assumption on the query of the (RQGS) model and draw query sequences according to probabilities given by an HMM.

In all these cases EVDs are still used heuristically. Hence, one attempts to fit the parameters of the EVD by straightforwardly sampling from the score distribution. This is, we generate pairs of random sequences according to the given null model and calculate the corresponding alignment score. Such an approach is e.g. implemented in the `hmmcalibrate` program from the HMMER package [19]. Nevertheless, this may fail to describe the tail of the distribution correctly, although this is most important for the estimation of statistical significance.

Our motivation for a simulation-based method that makes no initial parametric assumption refers to the approach [20] to increase the sensitivity of detecting homologs of a given transmembrane (TM) protein in a database search: A bipartite scoring scheme with a (non-symmetric) transmembrane helix specific scoring matrix (such as SLIM [20]) for the TM helices and a general-purpose scoring matrix (such as BLOSUM [6]) for the remaining regions of the query protein were applied, see Figure 1. This results in higher search sensitivity and specificity. However, a statistical theory or efficient computational method in such a (FQPS) framework is missing so far.

Our contributions and paper outline

We present a general framework (Algorithm 2) for efficient estimation of the tail of raw score distributions in sequence comparison problems. We only make the following assumptions:

1. We are able to sample sequences x according to the null model and to compute the null model

[1] The E-value for score s is the expected number of database hits with score at least s and depends on both $pval(s)$ and the database size.

probability of any given x.

2. We have an efficient algorithm $\mathcal{A}$ that computes the score $S(x, y)$, where x, y could be a pair of random sequences (RQGS or HMM), or one fixed and one random sequence (FQPS).

3. The scores are rational numbers with a common denominator. Hence, without loss of generality, they can be assumed to be integers.

4. Optionally for the (HMM) approach, we have an efficient algorithm $\mathcal{V}$ that predicts the most likely state sequence for a given sequence.

Our framework is readily applicable to the (RQGS), (FQPS) and (HMM) models, but also to more exotic settings, such as normalized alignment [4], where the score is not additive, but normalized by the alignment length, for which no statistical framework exists so far. Very recently Eddy [21] studied the distributions of Viterbi and Forward scores under probabilistic local alignment, for which a numerical analysis of the rare-event tail would be of interest as well.

In the current stage of the methodology, the computation of an accurate "on the fly" p-value for each particular database query might be impracticable as the convergence is not achieved within a few minutes. Therefore one might compromise between the i.i.d. assumption on the query and a more specific model to precompute score distributions. Our method is applicable to models where null probabilities can be computed efficiently, such as HMMs, normalized alignment, and even stochastic context-free grammars.

We will illustrate the approach for the HMM for TM proteins (TMHMM [22, 23]), which has been proven valuable in predicting TM helices. In this approach (and possible in other models as well) one is able to specify the score distribution in more detail, in the sense that a query might be classified into various sub-classes $\mathcal{C}$ with individual score distributions. A natural classification of the TMHMM is the number of transmembrane regions of the most likely prediction.

The rest of the paper is organized as follows. The following section presents the mathematical background on importance sampling and Markov chain Monte Carlo methods followed by a description of the methodology. 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 "Results" shows computational results on transmembrane protein similarity statistics in (RQGS), (FQPS) and (HMM), particularly the dependency of the score distribution $P(s)$ on the sub-classes $\mathcal{C}$ is studied. A discussion concludes the paper.

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 $\mathbb{E}[h(Z)]$, where Z is a random object and h is a real-valued function. We assume that we can draw n random samples $Z_1, \ldots, Z_n$ from the null model. The expectation is then approximated by the empirical mean $\mathbb{E}[h(Z)] \approx 1/n \cdot \sum_{i=1}^{n} h(Z_i)$.

In our setting, to estimate the score distribution (and then p-values), we consider the state space $\mathcal{Z} = \Sigma^{L_\mathtt{q}} \times \Sigma^{L_\mathtt{s}}$, from which we generate N random pairs of sequences $(\{X_1, Y_1\}, \ldots, \{X_N, Y_N\})$. These pairs are then aligned by a given algorithm $\mathcal{A}$ and the corresponding similarity scores $S(X_i, Y_i)$ are computed. We consider the family of functions $h_s : \mathcal{Z} \to \{0, 1\}$ for all $s > 0$, defined by $h_s(x, y) := 1$ if $S(x, y) = s$, and $h_s(x, y) := 0$ if $S(x, y) \neq s$. So

$$\text{Prob}(S(X, Y) = s) = \mathbb{E}[h_s(X, Y)] \approx |\{i : S(X_i, Y_i) = s\}|/N$$

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 afterward, which results in a more accurate estimate with a reasonable number of samples. Let p be the probability mass function (pmf) of (X, Y), and let q be another pmf satisfying $q(x, y) > 0$ whenever $p(x, y) > 0$. Then

$$\begin{aligned}
\mathbb{E}_p[h(X, Y)] &= \sum_{x,y} h(x, y) \cdot p(x, y) \\
&= \sum_{x,y} h(x, y) \cdot \frac{p(x, y)}{q(x, y)} \cdot q(x, y) \\
&= \mathbb{E}_q\left[h(X', Y')\frac{p(X', Y')}{q(X', Y')}\right] \\
&\approx \frac{1}{n}\sum_{i=1}^{n} h(X'_i, Y'_i) \cdot \frac{p(X'_i, Y'_i)}{q(X'_i, Y'_i)}, \qquad (1)
\end{aligned}$$

where each pair (X'_i, Y'_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; second, we need to be able to sample according to q; third, we need to be able to compute the correcting weight $p(x, y)/q(x, y)$. Since directly 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 no simple direct method to do so, the Metropolis-Hastings method [24] provides a solution by constructing an ergodic Markov chain with stationary distribution q in the following way.

Let us call the elements (X, Y) of the sample space $\mathcal{Z}$ *configurations*. Each configuration (x, y) has a set $\mathcal{N}(x, y)$ of potential neighbors $(x', y') \in \mathcal{N}(x, y)$ that are proposed with positive probability $P_{(x,y),(x',y')}$ as the next configuration. The proposal is accepted with probability

$$\alpha\left((x, y) \to (x', y')\right) = \min\left\{1, \frac{q(x', y') \cdot P_{(x',y'),(x,y)}}{q(x, y) \cdot P_{(x,y),(x',y')}}\right\}, \tag{2}$$

in which case (x', y') becomes the new configuration (x, y). Otherwise (x', y') is discarded and (x, y) remains unchanged.

For an appropriate choice of neighborhoods $\mathcal{N}(x, y)$ and of $P_{(x,y),(x',y')}$, 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, y) is sampled after equilibration, it will behave like 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. This time is usually referred as *mixing time*.

Methodology

The crucial point of the Metropolis-Hastings update (see Algorithm 1) is the choice of an appropriate neighborhood $\mathcal{N}(x, y)$ and the computation of the probabilities of newly proposed states $q(x', y')$. The neighborhood should be chosen such that the acceptance rate Eq. (2) is between 0.3 and 0.7. We shall factorize the (unnormalized) pmf q in two contributions, firstly weights $w : \mathbb{Z} \to \mathbb{R}^+$ that assign each score value of interest a weight and secondly the null probability, i.e.

$$q(x, y) = w(S(x, y)) \cdot p(x, y)\,. \tag{3}$$

Note that we will leave $w(\cdot)$ undetermined for a moment, until Section "Wang-Landau Sampling". The importance reweighting equation Eq. (1) for h_s is then

$$\mathrm{Prob}(S = s) = \mathbb{E}\left[h_s(X, Y)\right] = \sum_{x,y} h_s(x, y) \cdot p(x, y) \approx \frac{1}{Z} \sum_{i=1}^{N} \frac{h_s(X_i', Y_i')}{w(s)} \tag{4}$$

with the normalization constant $Z = \sum_s \sum_{i=1}^{N} \frac{h_s(X_i', Y_i')}{w(s)}$.

The (HMM) contains more information than the distribution of S in the sense that each query is a member of a certain sub-class characterized by the number of transmembrane regions "# of TM helices" to be determined by the Viterbi algorithm. Each class has its own probability

$P_n(s) = \text{Prob}(S = s | \#\text{ of TM helices} = n)$. In order to take this property into account, we deal with the joint probability $\text{Prob}(S = s, \#\text{ of TM helices} = n)$. Accordingly, the weights have a two dimensional domain $w : \mathbb{Z} \times [0, n_{\max}] \to \mathbb{R}$ and h_s in Eq. (4) is replaced by an indicator function $h_{s,n}$ that depends on two parameters.

Generally the occurrence of $x = x_1 \dots x_{L_\mathfrak{q}}$ and $y = y_1 \dots y_{L_\mathfrak{s}}$ is characterized by the null probability

$$p(x, y) = \text{Prob}(X = x, Y = y) = f^{\text{query}}(x_1 \dots x_{L_\mathfrak{q}}) \cdot f^{\text{subject}}(y_1 \dots y_{L_\mathfrak{s}}).$$

This simple factorization allows us to draw proposals for the query and for the subject independently, i.e. first one of the two sequences is chosen at random [2] with probability 1/2. Then one sequence of its neighbors is proposed as one partner of the new pair. Hence, we only need to consider the proposal densities $P_{x,x'}$ and $P_{y,y'}$. For the three different models under consideration these are specified as follows.

Proposal densities for (FQPS) and (RQGS)

In the simplest case either both sequences are i.i.d. or the query is fixed (to some sequence $\tilde{x}$) and the probabilities of their occurrence factorize, i.e.

$$f^{\text{query}}(x) = \begin{cases} f^{\text{iid}}(x) = \prod_{i=1}^{L_\mathfrak{q}} f_{x_i} & \text{for (RQGS) and} \\ \mathbf{1}_{\{x=\tilde{x}\}} & \text{for (FQPS)} \end{cases} \tag{5}$$

and of course $f^{\text{subject}}(y) = f^{\text{iid}}(y) = \prod_{i=1}^{L_\mathfrak{s}} f_{y_i}$ in both cases.

Due to the factorization that occurs in Eq. (5) it is possible to draw sequences from $\mathcal{N}(x)$ such that the detailed balance condition $f^{iid}(x) \cdot P_{x,x'} = f^{iid}(x') \cdot P_{x',x}$ is fulfilled by the following set of Monte Carlo moves (see also Figure 2 and Table 1)

a) substitution at position k,

b) insertion at position k with left shift,

c) insertion at position k with right shift,

d) deletion at position k with left shift,

[2] In the case of (FQPS) the subject is always chosen.

e) deletion at position k with right shift.

Operation a) appears with probability $1/2$ and the other ones with probability $1/2 \cdot 1/4$ each. This is one possible choice that guarantees detailed balance.

Note that all sequences in $\mathcal{N}(x)$ have the same length and each operation involves a replacement of an existing letter with a newly drawn letter, in case a) by a direct substitution and in the cases b)-e) indirect via a shift operation. Each position of a sequence has the same probability of being chosen and the replaced letter is chosen in all cases according to the probabilities f_σ ($\sigma \in \Sigma$).

With this construction the Metropolis-Hastings ratio Eq. (2) simplifies to the special case of the Metropolis algorithm, i.e.

$$\alpha\left((x,y) \to (x',y')\right) = \min\left\{1, \frac{w(S(x',y'))}{w(S(x,y))}\right\}, \tag{6}$$

where the acceptance rate depends on the score values only.

Proposal densities for the (HMM)

In contrast to the approach presented in the previous section, the generalized method we use here also works for null models that do not allow for direct sampling from $\mathcal{N}(x)$ as in the case of i.i.d. sequences. In principle this framework, summarized in Algorithm 1, is applicable to all models that allow for a rapid calculation of the null probabilities $f(\cdot)$. We first explain the ingredients before stating the algorithm.

Let us briefly state some important features of HMMs [2, 25]. In this general probabilistic framework one assumes that a sequence of observed symbols is generated through a sequence of "hidden" states. This state sequence, also called *path*, follows a simple Markov chain. The states are connected to the output symbols through emission probabilities; that is, a state can produce a symbol according to a distribution over all possible symbols. More formally, a HMM consists of

- a finite set Σ of symbols (in our case the amino acid alphabet),
- a finite set Γ of (hidden) states,
- inital state probabilities π_μ for all $\mu \in \Gamma$ with $\sum_{\mu\in\Gamma} \pi_\mu = 1$,
- emission probabilities p_σ^μ in each state $\mu \in \Gamma$ and for all $\sigma \in \Sigma$ with $\sum_{\sigma\in\Sigma} p_\sigma^\mu = 1$,
- a stochastic transition probability matrix $P = (p_{\mu,\tau})_{\mu,\tau\in\Gamma}$, i.e. $\sum_{\tau\in\Gamma} p_{\mu,\tau} = 1$.

Given these model parameters and a fixed sequence $x = x_1 \dots x_L$ of output symbols, the state sequence $Z = Z_1 \dots Z_L$ is a stochastic process.

For the Monte Carlo sampling as needed here, it is not possible to simulate a HMM directly to generate output sequences, since importance sampling changes the underlying sequence probabilities. Nevertheless, one still needs to compute the probabilities $f^{\mathrm{HMM}}(x)$ for the Monte Carlo acceptance procedure, i.e. the probabilities that x is the observed sequence generated by the HMM. These probabilities can be computed in $\mathcal{O}(L \cdot |\Gamma|^2)$ time using the well known *forward algorithm* as described in the following. One introduces the auxiliary variables $f_\mu(i)$, which correspond to the probability that the subsequence $x_1 \dots x_i$ is generated by the model given that the last state variable Z_i has the value μ, i.e. $f_\mu(i) = \mathrm{Prob}(X_1 \dots X_i = x_1 \dots x_i | Z_i = \mu)$. The overall probability is then $f^{\mathrm{HMM}}(x) = \sum_{\mu \in \Gamma} f_\mu(L)$. The probabilities $f_\mu(i)$ can be determined by the recursion

$$f_\mu(i) = p^\mu_{x_i} \sum_{\tau \in \Gamma} f_\tau(i-1)\, p_{\tau,\mu} \tag{7}$$

with initial conditions $f_\mu(1) = \pi_\mu p^\mu_{x_1}$.

Within the same time complexity the *Viterbi algorithm* $\mathcal{V}$ computes the most probable state path for a given sequence of observations, that is

$$z_1 \dots z_L = V(x_1 \dots x_L) = \operatorname*{argmax}_{\bar{z}_1 \dots \bar{z}_L \in \Gamma^L} \mathrm{Prob}(Z_1 \dots Z_L = \bar{z}_1 \dots \bar{z}_L | x_1 \dots x_L).$$

Let $v_\mu(i)$ be the probability of the most probable path ending in state $\mu \in \Gamma$ with observation x_i. These values can be computed recursively by

$$v_\mu(i) = p^\mu_{x_i} \max_{\tau \in \Gamma} \{ v_\tau(i-1)\, p_{\tau,\mu} \} \tag{8}$$

with boundary condition $v_\mu(1) = \pi(\mu) \cdot p^\mu_{x_1}$. Note that these probabilities are not normalized, in particular $\sum_{\mu \in \Gamma} v_\mu(i) \leq 1$. The most probable path is reconstructed by backtracking [25].

The HMM approach we use to sample transmembrane queries is the TMHMM developed by Sonnhammer et. al. [22]. In this setting, the output symbols are (structural) domains, and hidden states are "tied" according to their emission probabilities. They are classified into seven groups:

- Helix core,
- two different groups of caps on either side,
- loops on the cytoplasmic side,

- short and long loops on the non-cytoplasmic side,
- globular domains.

The internal structure of the helix core and loop module allows modeling different lengths of the corresponding protein domain by assigning jump probabilities. The globular domains have a self-looping structure and hence may also have various lengths. The other modules have fixed length. The overall number of model parameters is 216. Figure 3 shows the actual layout of TMHMM. Each box represents a group of "tied" states. The states corresponding to "helix core" represent the transmembrane helices that connect states of the cytoplasmic side and the non-cytoplasmic side of the membrane. The prediction of the positions of the "helix core" states determines the loci of the special purpose scoring matrix SLIM for position specific alignment (see Figure 1).

The following Metropolis-Hastings update (Algorithm 1) consists of two steps: First, the proposal of a new configuration from the neighborhood $\mathcal{N}(x)$ is made by inserting/replacing letters with equal weights $f_\sigma = \frac{1}{|\Sigma|}$ for all $\sigma \in \Sigma$ using one of the five Monte-Carlo moves described above. The acceptance ratio Eq. (2) in that case is given by

$$\alpha\left((x,y) \to (x',y')\right) = \min\left\{1, \frac{w(S(x',y')) \cdot f^{\text{query}}(x') \cdot f^{\text{subject}}(y')}{w(S(x,y)) \cdot f^{\text{query}}(x) \cdot f^{\text{subject}}(y)}\right\}. \tag{9}$$

The second step of the algorithm is based on the TMHMM. This allows us to sample non-i.i.d. sequences with appropriate weights and to predict transmembrane helical regions that can be used in the position specific alignment scheme (as described in [20]) even for random sequences.

Wang-Landau Sampling

The idea of importance sampling is to choose the weights $w(\cdot)$, such that the drawn events in the region of interest have a high probability to occur in the simulation. Ideally, $P(s)$ is already known and in that case one might choose $w(s) \propto 1/P(s)$ on the entire range of interest. Then all states are visited with equal probability, and hence a flat score histogram is achieved in the limit of infinite sample size. This idea refers back to statistical physics and it is known as "generalized ensemble" or "flat histogram" methods. In the following we will denote this weights by w^{flat}.

Of course the *true* $P(s)$ is unknown and the method requires some guesses which approximate w^{flat} to a suitable accuracy. The achieved score histogram becomes only approximatively flat. The true (unknown) distribution can then be estimated by reweighting the histogram of visited states using the importance sampling formula Eq. (4) for h_s.

Many iterative sampling schemes to achieve initial guesses had been developed in the 1990ies, for example entropic sampling [26], multicanonical sampling [27] and later transition matrix Monte Carlo [28–30], only to mention a few. Here we use the Wang-Landau algorithm [31,32] to approximate w^{flat} as input for Metropolis-Hastings sampling.

The Wang-Landau algorithm (Algorithm 2) explicitly violates detailed balance by dynamically updated weights depending on the visited states in the following way: First, a score range of interest $[s_{\min}, s_{\max}]$ is chosen. The algorithm basically employs a histogram $H(s)$ and weights $w(s)$ defined on the desired score range. For more complicated models such as the (TMHMM), these objects are two-dimensional depending on the score s and the class n, i.e. $H(s,n)$ and $w(s,n)$. Furthermore, real valued parameters $\phi_i > 1$ are used in each iteration i. Initially, the histogram values $H(s,n)$ are set to 0 in the desired range and all weights $w(s,n)$ to a constant, say 1. For the first iteration, $i = 0$, ϕ_i can be as large as e^1. Then, a simulation is performed using acceptance ratio Eq. (6) or Eq. (9). After each step, corresponding to one step of a (biased) *random walk* in configuration space, $w(s,n)$ is updated as $w(s,n) \leftarrow w(s,n)/\phi_i$, where s is the current score value and n the sub-class of the current state. Also the histogram H is updated by one $H(s,n) \leftarrow H(s,n) + 1$. This is continued until an "approximately flat histogram" is achieved. A possible criterion might be $H(s,n) > 0.6 \cdot \frac{1}{s_{\max}-s_{\min}+1} \sum_{s'=s_{\min}}^{s_{\max}} H(s',n)$ for all s, n. Once the histogram is "flat", ϕ is decreased by the rule $\phi_{i+1} \leftarrow \sqrt{\phi_i}$ and all entries of the histogram H are set to 0 again, while w is kept for the next iteration. Note that the flatness criterion is not essential for the algorithm. It is enough to guarantee that all values of s have been visited, for example by requiring that the random walker has cycled several times through the interval of interest $[s_{\min}, s_{\max}]$.

Due to the decreasing rule $\phi_{i+1} \leftarrow \sqrt{\phi_i}$, the modification factor ϕ converges towards 1. The simulation is stopped when ϕ reaches a value which is close to 1. It turned out that in our case the range from $\phi_0 = \exp(0.1) \approx 1.105$ to $\phi_{\text{final}} = \exp(0.0002) \approx 1.0002$ has been proven valuable.

Since detailed balance is violated explicitly, the convergence of the algorithm can not be proven. For this reason one should always perform a simulation with $\phi = 1$ for data production, which corresponds to the Metropolis-Hastings algorithm.

Improvements

Of course there is much room for improvement. For example, consider the time evolution of the histogram $H(s)$ for RQGS with $L_Q = L_S = 348$ up to $s_{\max} = 500$ with $\text{Prob}(S = s_{\max}) \approx 10^{-65}$ in Figure 4a. When starting with an initial guess $w(s) = 1$ for all $s \in [23, 500]$, the random walker needed about

5.8×10^5 Monte-Carlo steps for a *round trip*, i.e. to move from the lowest score $s_{\min} = 23$ to the highest one $s_{\max} = 600$ and back. The duration of a round trip is a measure of the mixing time of the corresponding Markov chain. Hence, the shorter the round trip time, the faster the convergence. During the first round trip, the weights have been improved such that the second round trip (and further round trips) needed only 13% of the computational effort of the first one. Once the random walker has performed its first round trip, the typical round trip time does not change significantly. This tight bottleneck in the very early stage of the algorithm can be overcome by suitable initial guesses of w. In Figure 4b the time evolution of the same parameter set (RQGS with $L_Q = L_S = 348$) is shown except for the choice of the weights, which have been chosen as $w(s) \approx 1/\mathrm{Prob}(S = s|L_Q = 348, L_S = 200)$, i.e. from a previous simulation of a different but similar setup. One observes that the histogram becomes "flat" within a much smaller amount of Monte-Carlo steps. Furthermore, the first round-trip time decreases to 1.3×10^5 (i.e. 22% of the value for the naive guess $w(s) = 1$). From the practical point this allows for saving computer time for simulations for a target parameter set B that is (more or less) close to a parameter set A for which a suitable approximation of $\mathrm{Prob}(S = s)$ has already been obtained. However, there remains the task to iterate the Wang-Landau algorithm down to values of ϕ that are close enough to 1. In cases where the two parameter sets A and B are sufficiently close to each other, in the sense that the score distributions $\mathrm{Prob}(S = s)$ do not differ too much, even that might be unnecessary. In that case it suffices to run a short batch run with $\phi = 1$, i.e. a detailed balance simulation, and then apply importance reweighting and use the so obtained approximation of $P(s)$ for a longer production run. This kind of procedure is shown in the inset of Figure 4b: The detailed balance simulations were performed with $L_Q = L_S = 348$, whereas the weights $w(s)$ came from a simulation with $L_S = 320$ and $L_S = 400$, respectively. The result shows that the histograms are not "flat" at all, but the distributions were close enough to visit all score values on the range of interest. In this successive way of iterations a broad range of the parameter space is accessible.

Estimation of the statistical error

Statistical analysis of Markov-chain Monte-Carlo data requires a careful inspection of correlation effects because the events depend on the history of the chain. This correlations vanish within a typical timescale and events that are separated by a sufficient number of steps can be assumed to be independent. However, since Monte-Carlo methods are only approximative, an assignment of statistical errors are requisite. In this study we used Flyvbjerg and Peterson's [33] blocking method to estimate the error.

Results

To our knowledge we present the first highly accurate score statistics for alignments with position-specific scoring schemes. The alignment scores were calculated with the standard Smith-Waterman algorithm with the BLOSUM62 matrix for the (RQGS) and a bipartite version BLOSUM62/SLIM for (FQPS) and (HMM) (see Figure 1). For the affine gap costs we have chosen the standard values with a gap-open penalty of 12 and a gap-extension penalty of 1, and UniProt letter frequencies for i.i.d. sequences.

We discuss four different transmembrane proteins as queries (see Table 2) in the (FQGS) scheme. The results are shown in Figure 5, where the distributions of (FQGS) and (RQGS) are compared against each other. The subject lengths are set to the query lengths. For the production run of one distribution in Figure 5 ($L_{\mathtt{Q}} = LS = 348$) 16,777,216 Metropolis-Hastings updates have been performed. This took about 16 hours on an Intel Pentium 4 with 3.4GHz. The performance of the corresponding HMM is weaker for three reasons: Firstly, we are interested in a joint distribution for that we need more samples. Secondly, more proposals are rejected from the sampler due to the HMM-weights and finally the computation of the forward-probabilities requires additional floating point operations. The computation of 16,777,216 Metropolis-Hastings updates for this model costs about 45 CPU hours. We use an 8 times larger sample size in order to account for the first drawback. Hence, we put an overall computational effort on this model, which is 23 times as large as for (FQGS) and (RQGS) (apart from the Wang-Landau iterations). Here we observe that the curvature is more pronounced in the (FQPS) model: Significant differences of shapes already show up in the high probability region, which is accessible by simple sampling (Figure 5a). All (RQGS) distributions match almost perfectly (only two lengths are shown), whereas the shape of the (FQPS) distributions varies slightly with the sequence type. This shows that position-specific scoring in connection with a fixed query sequence may better discriminate between different sequences than the standard approach of having two random sequences in connection with position-independent scoring, as already claimed in [20].

If the score distribution follows a Gumbel form $\mathrm{Prob}(S > s) = 1 - \exp[-c \cdot \exp(-\lambda s)]$, then, in the far right tail, essentially $\mathrm{Prob}(S > s) = c \cdot \exp(-\lambda s)$ since $1 - \exp[-\varepsilon] = \varepsilon$ (numerically) for very small $\varepsilon > 0$. Hence, $\mathrm{Prob}(S = s) = c' \cdot \exp(-\lambda s)$ (with $c' = c \cdot (e^{\lambda} - 1)$), and $\log \mathrm{Prob}(S = s)$ 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 [13], 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 5b are not straight lines in the right tail, but have negative curvature.

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

$$\log \mathrm{Prob}(S = s) = \log(\lambda) - \lambda\,(s - s_0) - \lambda_2\,(s - s_0)^2\,, \tag{10}$$

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 correction has been proposed for (RQGS) statistics and has been computed for different parameter sets of BLOSUM62 and PAM250 with affine gap costs [14, 15].

More pronounced differences are seen in the behavior of the tail (Figure 5b), 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. Note that for the (RQGS) model, previous simulations [15] have already shown that for the special case of $L_{\mathtt{s}} = L_{\mathtt{q}}$, the pmf converges for large sequence length indeed to an EVD.

In the rare-event tail, alignment lengths are in the order of the query length, hence the alignment is effectively global, or, in terms of statistical mechanics, optimal alignments "percolate". This is the same kind crossover that can be observed in the high-probability region when crossing the phase boundary of the linear-logarithmic transition [34] by decreasing the gap-costs. When the gap costs are small enough, the length of local alignments is also in the order of the query length even for random non-homologous sequences. This is referred as "linear phase".

Sardiu, Alves and Yu [35] studied the statistics of global alignments in the linear and logarithmic phase. They could classify the distribution to be either different types of the Tracy-Widom distribution [36] in the linear phase, or an exponential distribution in the logarithmic phase.

In the sense that those alignments that are responsible for the deviations from the Gumbel form can be seen as global alignments, it is rather a crossover to aforementioned Tracy-Widom distribution than to a "Gaussian" distribution as it was previously termed [14, 15]. For large score values this distribution scales with an exponent which is smaller than 2 (in the case of ($\beta = 2$)-Tracy-Widom distribution with

$P(s^*) \sim \exp[-{s^*}^{3/2}]$, [36]) with $s^* := s - s_0$. Hence we also tried to fit a modified Gumbel distribution with a correction term $\lambda_2 |s - s_0|^{3/2}$. This fits the data quite well, but we have obtained a much larger χ^2 value. For $L_{\mathtt{Q}} = L_{\mathtt{S}} = 400$ we observed an reduced χ^2 of ≈ 0.4 for an parabolic correction and ≈ 3 for the Tracy-Widom correction. The heuristic approximation Eq. (10) seems to be useful for practical purposes. This was further supported by a fit, where the exponent was considered as free parameter, i.e. $\lambda_2 |s - s_0|^{\gamma}$. For (RQGS) and $L_{\mathtt{Q}} = L_{\mathtt{S}} = 400$ we obtained $\gamma = 2$ within the errorbars, where the initial value for the fit procedure was set to $\gamma = 3/2$. Note that for a more direct comparison with Ref. [35], we would have to consider reparameterized alignment scores which would also imply different importance sampling distributions. Another integral difference to our work is that Sardui et. al. studied the statistics of a simplified alignment model where the score matrix is modeled by a single probability.

The modified Gumbel statistics affect a possible ranking of database search results only slightly (it does not change if both sequences were of equal length). Nevertheless, the threshold of significant hits is shifted towards lower scores, especially if one is interested in intermediately strong homological relationships. To illustrate this, we used BLAST to receive homologs of our four example proteins from the current Swissprot database. We considered the first 1000 hits, ignoring all results with $L_{\mathtt{S}} > 800$ because this was beyond the lengths under consideration in the simulations. The scores were recomputed via the Smith-Waterman algorithm for (RQGS) and via the position specific Smith-Waterman algorithm for (FQPS), and we computed the corresponding p-values from our empirical data. For subject sequence lengths that are not directly governed by our simulation directly we used interpolated fit parameters. Next, we considered the number of hits below E-value thresholds based on three statistics, the BLAST statistics, the statistics of the (RQGS) model and finally the model of (FQPS). We treat the E-value as the product of number of entries in the database times p-value. The results are shown in Figure 6. In the search of homologs with relatively small E-value thresholds of $\sim 10^{-15}$ classical methods such as the compositional adjustment as it is implemented in the BLAST algorithm, are sensitive enough to discriminate significant hits against random ones. However, in applications one possibly wish to refine a BLAST result set by filtering sequences below a certain critical E-value. If this value is in the order of magnitude 10^{-15} (or below) one would possibly miss relevant hits when using the classical statistics. Therefore we suggest to use the rare-event statistics after retrieving the data. This yields an accurate threshold value which can be used as further filter criterion. When the original BLAST statistics is used for that purpose one would possibly miss a large fraction of hits. We also detected a clear difference between the models (FQPS) and (RQGS) could be detected. With (RQGS) for transmembrane proteins

one would possibly miss some hits during post-processing.

To investigate the impact of dissimilar 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 $L_{\mathtt{S}}/L_{\mathtt{Q}}$ (see Figure 7). The 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 the (RQGS) model, 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 correction term 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.

For the (HMM), we approximate the score distribution within each class (number of helices= n). The shape of the distributions clearly agrees with the curvature for (RQGS) and (FQPS), and the modified Gumbel distribution could be fitted (see Figure 8) when the number of helices was not too small. This is indicated by a large reduced χ^2 value for distributions with a small number of helices. Also a visual inspection of the fit to the data supports this argument.

The rare-event tail shows clear differences between the different sub-classes of the model over several orders of magnitude. In Figure 9 the dependency of the fit parameters on the respective sub-class of the model (Figure 9a and Figure 9b) as well as the dependency on the ratio $L_{\mathtt{S}}/L_{\mathtt{Q}}$ (Figure 9c and Figure 9d) is shown. Note that for distributions that are not well described via Eq. (10), we only fitted the data in the high probability region. Those data points are left out in the plot for λ_2 in Figure 9b and are connected by dotted lines in Figure 9a.

In analogy to (RQGS) and (FQPS), the curvature remains constant when $L_{\mathtt{S}} > L_{\mathtt{Q}}$. Regarding the dependence on the number of helices, the curvature decays with increasing number of transmembrane regions and then approaches an approximate constant value. Numerical values are provided in the Appendix for reference.

Concluding Discussion

We have presented a simple universal method to accurately sample the far right tail of the score distribution of various sequence comparison algorithms. It appears to be the first method that is applicable to all 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 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, generating the 16,777,216 samples for Figure 5 ($L_{\mathtt{Q}} = L_{\mathtt{S}} = 348$) took approximately 16 hours on an Intel Pentium 4 with 3.4GHz.

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 weights in the generalized ensemble [37, 38].

While this still prohibits interactive use, we see a lot of potential for our method to provide an improved version of the `hmmcalibrate` tool [19] and to explore the statistics of normalized sequence alignment [4].

During the preparation of this manuscript we came aware of a new related importance sampling method which is suitable for efficient p-value computations for alignment statistics [39]. So far this method was applied to i.i.d. sequences but it should be possible to extend it to more complex model as well.

Authors contributions

SW developed the simulation program for (FQPS) and HMM based on an earlier version [15], ran the simulations and performed data analysis. SR and AKH designed the project. IH and SW developed the details for the TMHMM. All authors contributed to the manuscript.

Acknowledgments

SW is supported by the German *VolkswagenStiftung* (program "Nachwuchsgruppen an Universitäten"). IH is supported by the NRW Graduate School in Bioinformatics and Genome Research, Bielefeld. The simulations were performed at the "Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen" and at the cluster of the Center for Biotechnology at the university of Bielefeld. Allocation of computing time is greatfully acknowledged.

References

1. Smith TF, Waterman MS: **Identification of Common Molecular Subsequences**. *J.mol.Biol.* 1981, **147**:195–197.

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

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

4. Arslan AN, Egecioglu O, Pevzner PA: **A new approach to sequence comparison: normalized sequence alignment.** *Bioinformatics* 2001, **17**(4):327–337.

5. Bairoch A, Apweiler R, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N, Yeh LSL: **The Universal Protein Resource (UniProt).** *Nucleic Acids Res* 2005, **33**(Database issue):D154–D159, [http://dx.doi.org/10.1093/nar/gki070].

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

7. Mercier S, Daudin JJ: **Exact distribution for the local score of one i.i.d. random sequence.** *J Comput Biol* 2001, **8**(4):373–380, [http://dx.doi.org/10.1089/106652701752236197].

8. Karlin S, Altschul S: **Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes**. *Proc.Natl.Acad.Sci.U.S.A.* 1990, **87**:2264.

9. Gumbel E: *Statistics of Extremes*. New York: Columbia University Press 1958.

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

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

12. Altschul SF, Bundschuh R, Olsen R, Hwa T: **The estimation of statistical parameters for local alignment score distributions.** *Nucleic Acids Res* 2001, **29**(2):351–361.

13. Altschul S, Gish W: **Local Alignment Statistics**. *Meth. Enzym.* 1996, **266**:460.

14. Hartmann A: **Sampling rare events: Statistics of local sequence alignments**. *Phys. Rev. E* 2002, **65**:056102.

15. Wolfsheimer, S and Burghardt, B and Hartmann, A K: **Local sequence alignments statistics: deviations from Gumbel statistics in the rare-event tail**. *Algor. Mol. Biol.* 2007, **2**:9, [http://www.almob.org/content/2/1/9].

16. Yu YK, Wootton JC, Altschul SF: **The compositional adjustment of amino acid substitution matrices.** *Proc Natl Acad Sci U S A* 2003, **100**(26):15688–15693, [http://dx.doi.org/10.1073/pnas.2533904100].

17. Yu YK, Altschul SF: **The construction of amino acid substitution matrices for the comparison of proteins with non-standard compositions.** *Bioinformatics* 2005, **21**(7):902–911, [http://dx.doi.org/10.1093/bioinformatics/bti070].

18. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.** *Nucleic Acids Res* 1997, **25**(17):3389–3402.

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

20. Müller T, Rahmann S, Rehmsmeier M: **Non-symmetric score matrices and the detection of homologous transmembrane proteins**. *Bioinformatics* 2001, **17**:182–189, [http://bioinformatics.oxfordjournals.org/cgi/content/abstrac%t/17/suppl_1/S182].

21. Eddy SR: **A Probabilistic Model of Local Sequence Alignment That Simplifies Statistical Significance Estimation**. *PLoS Comput Biol* 2008, **4**(5):s1000069, [http://dx.doi.org/10.1371%2Fjournal.pcbi.1000069].

22. Sonnhammer EL, von Heijne G, Krogh A: **A hidden Markov model for predicting transmembrane helices in protein sequences**. In *Proc. Sixth Int. Conf. on Intelligent Systems for Molecular Biology.* Edited by et al JG, AAAI Press 1998:175–182.

23. Krogh A, Larsson B, von Heijne G, Sonnhammer ELL: **Predicting transmembrane protein topology with a hidden markov model: application to complete genomes**. *J. Mol. Biol.* 2001, **305**(3):567–580, [http://www.sciencedirect.com/science/article/B6WK7-457D7V9-K%/2/0367078014042718f39416a2c3ddeeb3].
24. Hastings WK: **Monte Carlo Sampling Methods Using Markov Chains and Their Applications**. *Biometrika* 1970, **57**:97–109.
25. R Durbin AK S Eddy, Mitchison G: *Biological Sequence Analysis*. Cambridge University Press 1998.
26. Lee J: **New Monte Carlo algorithm: Entropic sampling**. *Phys. Rev. Lett.* 1993, **71**(2):211–214.
27. Berg BA, Neuhaus T: **Multicanonical ensemble: A new approach to simulate first-order phase transitions**. *Phys.Rev.Lett.* 1992, **68**:9.
28. Wang JS, Tay TK, Swendsen RH: **Transition Matrix Monte Carlo Reweighting and Dynamics**. *Phys. Rev. Lett.* 1999, **82**(3):476–479.
29. Wang JS: **Transition matrix Monte Carlo method**. *Comput. Phys. Commun.* 1999, **121-122**:22–25, [http://www.sciencedirect.com/science/article/B6TJ5-3Y0HM2T-T%/2/3377e3546795e04c63dc23b6982b7459].
30. Wang JS, Lee LW: **Monte Carlo algorithms based on the number of potential moves**. *Comput. Phys. Commun.* 2000, **127**:131–136, [http://www.sciencedirect.com/science/article/B6TJ5-404H3KN-N%/2/e62d53facfd5d82de4b029380ea99a78].
31. Wang FG, Landau DP: **Efficient, multiple-range random walk algorithm to calculate the density of states**. *Phys. Rev. Lett.* 2001, **86**:2050.
32. Wang FG, Landau DP: **Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram**. *Phys. Rev. E* 2001, **64**:056101.
33. Flyvbjerg H, Petersen HG: **Error estimates on averages of correlated data**. *The Journal of Chemical Physics* 1989, **91**:461–466, [http://link.aip.org/link/?JCP/91/461/1].
34. Arratia R, Waterman M: **A Phase Transition for the Score in Matching Random Sequences Allowing Deletions**. *Ann.Appl. Prob.* 1994, **4**:200–225.
35. ME Sardiu and G Alves and Y Yu: **Score statistics of global sequence alignment from the energy distribution of a modified directed polymer and directed percolation problem**. *Phys.Rev.E.* 2005, **72**:061917.
36. Tracy CA, Widom H: **On orthogonal and symplectic matrix ensembles**. *Communications in Mathematical Physics* 1996, **177**(3):727–754, [http://dx.doi.org/10.1007/BF02099545].
37. Dayal P, Trebst S, Wessel S, Würtz D, Troyer M, Sabhapandit S, Coppersmith SN: **Performance Limitations of Flat-Histogram Methods**. *Phys. Rev. Lett.* 2004, **92**(9):097201–4, [http://link.aps.org/abstract/PRL/v92/e097201].
38. Trebst S, Huse DA, Troyer M: **Optimizing the ensemble for equilibration in broad-histogram Monte Carlo simulations**. *Phys. Rev. E* 2004, **70**(4):046701, [http://link.aps.org/abstract/PRE/v70/e046701].
39. Newberg LA: **Significance of Gapped Sequence Alignments**. *Journal of Computational Biology* 2008, **15**(9):1187–1194, [http://www.liebertonline.com/doi/abs/10.1089/cmb.2008.0125]. [PMID: 18973434].

Appendix: Modified Gumbel Parameters

Table 3 and Table 4 show numerical values for the parameters λ, λ_2 and K of the modified Gumbel distribution Eq. (10). These are visualized in Figures 7 and 9 in the body of the paper.

Figures

Figure 1 - Bipartite scoring scheme

Bipartite scoring scheme for the detection of homologous transmembrane proteins from Ref. [20]. 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 or generated by the TMHMM and the subject is assumed a random i.i.d. sequence drawn from the distribution of amino-acid frequencies of the database.

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.

Figure 3 - The layout of the HMM for transmembrane proteins

The layout of the HMM for transmembrane proteins according to Sonnhammer et.al. [22]. Each box corresponds to a group of states. For example the helix-core block consists of 25 internal states. Line type of boxes represent different emission probabilities. For more details we refer the reader to the original publication.

Figure 4 - Dynamics of the Wang-Landau algorithm

Typical time evolution of the histogram of visited states when starting with different initial guesses. The model parameters are RQGS with $L_Q = L_S = 348$. The weights have been updated dynamically with modification factor $\phi = \exp(0.1) \approx 1.105$. (a) $w(s) = 1$ for all s. The Markov chain converges relatively slowly. (b) $w(s) \approx 1/\text{Prob}(S = s|L_Q = 348, L_S = 200)$ has been used as an initial guess. The histogram becomes flatter within remarkable less computational effort. Inset: a detailed balance simulation ($\phi = 1$ during the simulation of $1,048,576$ steps) with initial weights that are close to the inverse target distribution. Though the histograms are not "flat", each score value on the interval $[23, 500]$ has been visited. The estimate from this data can be used in a longer production run.

Figure 5 - Score distributions for (RQGS) and (FQPS) models

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. (a): 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.

(b): 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.

Figure 6 - Number of hits as a function of the threshold

Number of BLAST hits as a function of E-value threshold considering different statistics: the original BLAST statistics as well as (RQGS) and (FQPS) statistics. The E-value of the rare-event statistics was approximated by the p-value times database size in terms of number of sequences.

Figure 7 - Fit parameters for (RQGS) and (FQPS) models

Dependence of the modified Gumbel parameters on the subject/query length ratio $L_{\mathtt{S}}/L_{\mathtt{Q}}$. The vertical line corresponds to Figure 5, where $L_{\mathtt{S}} = L_{\mathtt{Q}}$. (a): λ describes the bulk of the distribution (see Figure 5a) left). For $L_{\mathtt{S}} > L_{\mathtt{Q}}$, λ varies only slightly in the subject length.
(b): The parameter λ_2 characterizes the curvature of the pmf in the tail (see Figure 5b). 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 $L_{\mathtt{S}} > L_{\mathtt{Q}}$.

Figure 8 - Score distributions for different alignment models

Score distributions for different alignment models (i.i.d., fixed query and TMHMM) with $L_{\mathtt{S}} = L_{\mathtt{Q}} = 348$. The distributions for the (HMM) have been obtained from the joint distribution.

Figure 9 - Fit parameters for different alignment models

Fit parameters for score distributions $P(S|\#$ of helices) for the (HMM) with a fixed query length $L_Q = 348$ and various subject lengths L_S.
Both shape parameters λ and λ_2 decrease with increasing number of helices. The dependency on the subject length is stronger for λ_2 than for λ. For $L_S > L_Q$ the dependency of λ_2 on the subject length is only of marginal order. The bars show the distribution of the number of transmembrane helices obtained by direct simulations of the (HMM).
(c),(d): The L_S/L_Q dependency of λ and λ_2 extracted from the same data as (a),(b). The lines are guide to the eyes only. Dashed lines show the corresponding scaling behavior for the (FQRS) and (RQGS) models. The result for $n = 2$, that has been obtained from the high probability regions (see text), is indicated by dotted lines.

Algorithms

Algorithm 1 - Metropolis-Hastings update

MetropolisHastingsUpdate(x, y, z, p, s, n, w)

Input: Sequences x, y, a hidden state sequence z,

the null probability $p(x, y) = f^{\text{query}}(x) \cdot f^{\text{subject}}(y)$, the score $s = S(x, y)$, the sub-class n and weights w

Output: Possibly new values for x,y,z,p,s,n.

1: Draw $(x', y') \in N(x, y)$

2: compute $z' := V(x)$ using $\mathcal{V}$ and determine the corresponding class n';

3: compute $p' := f^{\text{query}}(x') \cdot f^{\text{subject}}(y')$;

4: compute $s' := S(x', y')$ using $\mathcal{A}$.

5: Compute $\alpha := \frac{w[s',n'] \cdot p' \cdot P_{x',x}}{w[s,n] \cdot p \cdot P_{x,x'}}$. ▷ *Designed such that* $p(x', y') \cdot P_{(x',y'),(x,y)} = p(x, y) \cdot P_{(x,y),(x',y')}$

6: With probability $\min\{1, \alpha\}$: Let $(x, y, z, p, s, n) \leftarrow (x', y', z', p', s', n')$

7: **return** (x, y, z, p, s, n)

The general Metropolis-Hastings update. It is assumed that the following are available: a sequence space $\mathcal{X}$, null distributions $f^{\text{query}}(\cdot)$ and $f^{\text{subject}}(\cdot)$ with efficient computation of $f^{\text{query}}(x)$ and $f^{\text{subject}}(y)$ for any $x, y \in \mathcal{X}$, neighborhoods $\mathcal{N}(x, y)$ with proposal distributions $P_{(x,y),(x',y')}$ for each x, y and a way to draw samples from $P_{(x,y),\cdot}$, a scoring algorithm $\mathcal{A} : \mathcal{X} \times \mathcal{X} \to \mathbb{Z}$ with $x, y \mapsto S(x, y)$ and optionally a Viterbi-like algorithm $\mathcal{V}$ that determines the most probable path and assigns any $x \in \mathcal{X}$ to a certain class n.

Algorithm 2 - Wang-Landau sampling

WangLandau$(w, \phi, \phi_{\text{final}}, N)$

Input: Initial guess $w[s, n]$, initial and final modification factors $\phi, \phi_{\text{final}}$, number of samples for production run N

Output: Histogram of visited scores, $H(s, n)$:= number of samples with score s and class n and weights used in the production run $w(s, n)$ for all s and n

1: ▷ *Initialize and estimate* $w[s, n]$

2: Pick any $x, y \in \mathcal{X}$ and compute its null probability $p := f^{\text{query}}(x) \cdot f^{\text{subject}}(y)$;

3: compute $s := S(x, y)$ using $\mathcal{A}$;

4: compute $z := V(x)$ using $\mathcal{V}$ and determine corresponding class n;

5: **while** $\phi > \phi_{\text{final}}$ **do**

6: $H[s', n'] \leftarrow 0$ for all possible score values s' and classes n'

7: **while** $H[s', n']$ is not flat **do**

8: $(x, y, z, p, s, n) \leftarrow$ MetropolisHastingsUpdate(x, y, z, p, s, n, w);

9: $H[s, n] \leftarrow H[s, n] + 1$; $w[s, n] \leftarrow w[s, n]/\phi$;

10: **end while**

11: $\phi \leftarrow \sqrt{\phi}$

12: **end while**

13: ▷ *Obtain N samples from q and their score counts / histogram*

14: $H[s', n'] \leftarrow 0$ for all possible score values s' and classes n'

15: **for** $i = 1..N$ **do**

16: $H[s, n] \leftarrow H[s, n] + 1$

17: **repeat**

18: $(x, y, z, p, s, n) \leftarrow$ MetropolisHastingsUpdate(x, y, z, p, s, n, w);

19: **until** mixing has occurred

20: **end for**

21: **return** counts H, weights w.

General sequence similarity score sampling framework. For general requirements see also Algorithm 1.

Tables

Table 1 - Monte Carlo operations

operation	resulting sequence
substitution of `D` at position 5	`LGQIDTAE`
insertion of `D` at position 5 with left shift	`GQIWDTAE`
insertion of `D` at position 5 with right shift	`LGQIDWTA`
deletion at position 5 with left shift	`LGQITAED`
deletion at position 5 with right shift	`DLGQITAE`

Valid Monte Carlo operations for input sequence $s =$ `LGQIWTAE` (indexing starts with 1). In order to obtain sequences of the same length as s, in the case of a deletion a character (D) to be appended at the border has to be specified.

Table 2 - A selection of transmembrane proteins

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

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

Table 3 - Fit parameters for (FQPS) and (RQGS)

		FQPS			corresponding RQGS		
L_Q	L_S	λ	$10^4\lambda_2$	K	λ	$10^4\lambda_2$	K
P08100	50				0.3016 ±0.40%	7.5741 ±0.77%	0.0654 ±3.34%
348	100	0.1747 ±0.19%	3.2202 ±0.32%	0.0132 ±1.49%	0.2829 ±0.17%	3.6884 ±0.36%	0.0463 ±4.09%
	200	0.1617 ±0.09%	1.7968 ±0.18%	0.0100 ±1.31%	0.2685 ±0.15%	1.8498 ±0.40%	0.0315 ±2.77%
	300	0.1478 ±0.14%	1.3962 ±0.21%	0.0059 ±2.20%	0.2664 ±0.14%	1.1900 ±0.47%	0.0292 ±3.49%
	320	0.1466 ±0.15%	1.3775 ±0.28%	0.0056 ±2.33%	0.2674 ±0.11%	1.1059 ±0.51%	0.0295 ±2.05%
	348	0.1432 ±0.22%	1.4131 ±0.33%	0.0051 ±2.69%	0.2681 ±0.10%	0.9909 ±0.43%	0.0307 ±2.18%
	360	0.1426 ±0.17%	1.4322 ±0.22%	0.0047 ±3.17%	0.2678 ±0.10%	0.9883 ±0.42%	0.0302 ±2.49%
	400	0.1418 ±0.10%	1.4201 ±0.17%	0.0047 ±1.43%	0.2648 ±0.12%	1.0238 ±0.50%	0.0248 ±3.89%
	500	0.1399 ±0.26%	1.4517 ±0.35%	0.0043 ±3.94%	0.2638 ±0.17%	1.0248 ±0.65%	0.0255 ±5.65%
	600	0.1405 ±0.16%	1.4392 ±0.20%	0.0047 ±2.87%	0.2650 ±0.14%	0.9917 ±0.74%	0.0245 ±3.85%
P50052	50				0.3024 ±0.85%	7.4294 ±1.70%	0.0657 ±6.19%
363	100	0.1795 ±0.16%	3.1869 ±0.26%	0.0132 ±1.42%	0.2818 ±0.25%	3.6993 ±0.55%	0.0458 ±3.44%
	200	0.1660 ±0.18%	1.8701 ±0.30%	0.0096 ±1.98%	0.2698 ±0.21%	1.8027 ±0.58%	0.0341 ±4.60%
	300	0.1550 ±0.22%	1.3995 ±0.36%	0.0066 ±2.97%	0.2643 ±0.14%	1.2232 ±0.42%	0.0273 ±3.55%
	330	0.1512 ±0.12%	1.4130 ±0.23%	0.0057 ±1.30%	0.2654 ±0.18%	1.0822 ±0.68%	0.0274 ±5.32%
	363	0.1509 ±0.18%	1.3881 ±0.27%	0.0057 ±3.53%	0.2687 ±0.24%	0.9676 ±1.00%	0.0332 ±7.75%
	380	0.1489 ±0.12%	1.4138 ±0.19%	0.0051 ±1.17%	0.2651 ±0.30%	0.9806 ±1.28%	0.0270 ±11.76%
	400	0.1474 ±0.20%	1.4335 ±0.32%	0.0048 ±3.27%	0.2634 ±0.15%	0.9773 ±0.75%	0.0271 ±11.41%
	500	0.1471 ±0.08%	1.4350 ±0.16%	0.0049 ±1.13%	0.2613 ±0.21%	0.9998 ±1.05%	0.0226 ±7.60%
	600	0.1457 ±0.28%	1.4640 ±0.54%	0.0046 ±3.24%	0.2662 ±0.15%	0.9498 ±0.79%	0.0250 ±7.76%
Q18179	50				0.3008 ±0.70%	7.6673 ±1.23%	0.0625 ±5.34%
455	100	0.1798 ±0.33%	3.7190 ±0.59%	0.0103 ±2.84%	0.2845 ±0.16%	3.5814 ±0.35%	0.0485 ±2.86%
	200	0.1723 ±0.16%	1.9839 ±0.32%	0.0087 ±1.50%	0.2685 ±0.14%	1.8391 ±0.49%	0.0302 ±3.81%
	300	0.1609 ±0.25%	1.4302 ±0.40%	0.0059 ±4.49%	0.2632 ±0.16%	1.2382 ±0.53%	0.0262 ±4.69%
	420	0.1569 ±0.27%	1.3665 ±0.52%	0.0050 ±2.90%	0.2636 ±0.17%	0.8441 ±0.59%	0.0222 ±9.17%
	450	0.1590 ±0.25%	1.3225 ±0.61%	0.0052 ±2.86%	0.2611 ±0.13%	0.8203 ±0.43%	0.0209 ±4.93%
	455	0.1548 ±0.26%	1.4038 ±0.52%	0.0049 ±2.76%	0.2655 ±0.12%	0.7670 ±0.49%	0.0246 ±8.35%
	480	0.1557 ±0.38%	1.3664 ±0.67%	0.0051 ±7.10%	0.2610 ±0.10%	0.7929 ±0.41%	0.0197 ±6.70%
	500	0.1521 ±0.45%	1.4145 ±0.77%	0.0044 ±5.30%	0.2615 ±0.17%	0.7783 ±0.62%	0.0204 ±5.09%
	600	0.1540 ±0.25%	1.3886 ±0.43%	0.0043 ±3.72%	0.2596 ±0.14%	0.7706 ±0.60%	0.0174 ±5.71%
P35348	50				0.3046 ±0.61%	7.3443 ±1.17%	0.0668 ±4.85%
466	100	0.1809 ±0.18%	3.1996 ±0.28%	0.0135 ±2.06%	0.2839 ±0.22%	3.6314 ±0.49%	0.0465 ±2.49%
	200	0.1625 ±0.12%	1.8687 ±0.18%	0.0079 ±1.63%	0.2696 ±0.15%	1.8030 ±0.48%	0.0315 ±3.97%
	300	0.1643 ±0.10%	1.2089 ±0.15%	0.0086 ±2.23%	0.2620 ±0.13%	1.2472 ±0.47%	0.0241 ±5.52%
	400	0.1510 ±0.24%	1.2641 ±0.39%	0.0051 ±2.76%			
	450	0.1521 ±0.33%	1.2357 ±0.55%	0.0050 ±5.39%	0.2647 ±0.16%	0.7874 ±0.67%	0.0246 ±3.93%
	466	0.1485 ±0.17%	1.2982 ±0.35%	0.0046 ±2.93%			
	480	0.1517 ±0.23%	1.2359 ±0.34%	0.0056 ±5.27%	0.2609 ±0.25%	0.7981 ±1.25%	0.0207 ±9.36%
	500	0.1492 ±0.22%	1.2845 ±0.35%	0.0048 ±3.64%	0.2668 ±0.09%	0.7124 ±0.49%	0.0265 ±6.00%
	600	0.1509 ±0.28%	1.2383 ±0.40%	0.0050 ±3.86%			

Fit parameters λ, λ_2 and K of the modified Gumbel distribution for (FQPS) and (RQGS).

Table 4 - Fit parameters for (FQPS) and (RQGS)

		HMM n=0			HMM n=1		
L_Q	L_S	λ	$10^4\lambda_2$	10^3K	λ	$10^4\lambda_2$	10^3K
348	150	0.2890 ±0.85%		49.4722 ±7.27%	0.2310 ±9.32%		21.4600 ±66.56%
	200	0.2894 ±2.84%		50.0796 ±24.47%	0.2274 ±1.74%		20.1017 ±13.25%
	300	0.2895 ±2.69%		53.3472 ±24.00%	0.2240 ±4.86%		17.8934 ±37.22%
	348	0.2988 ±3.24%		72.2356 ±30.15%	0.2234 ±2.39%		16.8704 ±18.79%
	360	0.2895 ±1.79%		51.9056 ±16.04%	0.2220 ±2.14%		16.3757 ±16.52%
	400	0.2859 ±3.49%		48.4496 ±31.10%	0.2232 ±2.40%		17.5141 ±18.94%
	500	0.2912 ±6.63%		54.0687 ±61.22%	0.2182 ±2.39%		14.7371 ±19.10%
	600	0.2901 ±3.38%		51.9412 ±31.74%	0.2180 ±2.59%		14.2439 ±20.86%
		HMM n=2			HMM n=3		
L_Q	L_S	λ	$10^4\lambda_2$	K	λ	$10^4\lambda_2$	K
348	150	0.1968 ±0.70%	2.9247 ±1.37%	12.0400 ±6.48%	0.1767 ±0.44%	2.6797 ±1.01%	7.4435 ±3.72%
	200	0.1947 ±2.12%		9.8704 ±14.29%	0.1795 ±0.46%	2.3586 ±0.92%	8.5733 ±3.87%
	300	0.1937 ±3.60%		9.9597 ±25.32%	0.1863 ±0.41%	2.0008 ±0.94%	11.7859 ±5.63%
	348	0.1888 ±3.19%		8.1338 ±22.42%	0.1876 ±0.32%	1.9328 ±0.89%	12.1223 ±3.83%
	360	0.1926 ±3.17%		9.7957 ±22.82%	0.1853 ±0.27%	1.9530 ±0.65%	10.8640 ±2.65%
	400	0.1934 ±1.05%		9.9321 ±8.22%	0.1757 ±1.64%		7.1756 ±11.58%
	500	0.1919 ±1.61%		9.3630 ±12.32%	0.1783 ±0.98%		7.7945 ±7.18%
	600	0.1912 ±1.70%		9.3303 ±13.25%	0.1768 ±1.01%		7.4165 ±8.19%
		HMM n=4			HMM n=5		
L_Q	L_S	λ	$10^4\lambda_2$	10^3K	λ	$10^4\lambda_2$	10^3K
348	150	0.1732 ±0.47%	2.2119 ±1.14%	7.4991 ±6.08%	0.1710 ±0.38%	2.0698 ±0.92%	8.1950 ±3.70%
	200	0.1686 ±0.28%	2.1187 ±0.72%	6.4162 ±3.14%	0.1657 ±0.39%	1.8231 ±1.14%	6.9148 ±3.82%
	300	0.1682 ±0.36%	1.9635 ±0.79%	6.5436 ±4.22%	0.1599 ±0.37%	1.7836 ±0.79%	5.4451 ±3.85%
	348	0.1685 ±0.35%	1.9408 ±0.74%	7.3851 ±3.34%	0.1580 ±0.28%	1.7930 ±0.68%	5.3049 ±2.61%
	360	0.1678 ±0.42%	1.9421 ±0.92%	6.5775 ±4.07%	0.1605 ±0.23%	1.7481 ±0.50%	5.7512 ±2.89%
	400	0.1662 ±0.18%	1.9782 ±0.40%	6.4164 ±2.32%	0.1587 ±0.28%	1.7828 ±0.73%	5.4513 ±2.57%
	500	0.1693 ±0.24%	1.9047 ±0.51%	7.0735 ±2.11%	0.1587 ±0.16%	1.7957 ±0.40%	5.4770 ±2.31%
	600	0.1693 ±0.17%	1.8994 ±0.39%	7.1112 ±2.06%	0.1575 ±0.29%	1.8330 ±0.58%	5.2125 ±2.68%
		HMM n=6			HMM n=7		
L_Q	L_S	λ	$10^4\lambda_2$	10^3K	λ	$10^4\lambda_2$	10^3K
348	150	0.1663 ±0.49%	2.1403 ±1.04%	7.9392 ±5.83%	0.1646 ±0.30%	2.1396 ±0.65%	8.7088 ±4.21%
	200	0.1614 ±0.25%	1.7767 ±0.65%	6.7568 ±2.30%	0.1574 ±0.41%	1.7687 ±1.17%	6.5219 ±3.81%
	300	0.1551 ±0.28%	1.5986 ±0.80%	5.2551 ±3.18%	0.1514 ±0.26%	1.4638 ±0.62%	5.0238 ±4.34%
	348	0.1531 ±0.20%	1.5993 ±0.55%	4.9132 ±2.71%	0.1482 ±0.33%	1.4755 ±0.77%	4.4535 ±4.13%
	360	0.1536 ±0.34%	1.6036 ±1.02%	4.9160 ±3.41%	0.1490 ±0.39%	1.4479 ±0.93%	4.6858 ±3.28%
	400	0.1537 ±0.27%	1.5713 ±0.62%	4.9524 ±3.05%	0.1494 ±0.24%	1.4328 ±0.70%	4.6867 ±2.08%
	500	0.1519 ±0.23%	1.6229 ±0.67%	4.6812 ±2.14%	0.1472 ±0.29%	1.4706 ±0.63%	4.2881 ±2.50%
	600	0.1489 ±0.15%	1.7148 ±0.33%	4.2283 ±2.16%	0.1460 ±0.18%	1.5193 ±0.49%	4.2679 ±1.74%
		HMM n=8			HMM n=9		
L_Q	L_S	λ	$10^4\lambda_2$	10^3K	λ	$10^4\lambda_2$	10^3K
348	150	0.1595 ±0.47%	2.2162 ±1.01%	7.5355 ±4.01%	0.1603 ±0.23%	2.1517 ±0.48%	8.0273 ±2.17%
	200	0.1534 ±0.55%	1.8019 ±1.46%	5.9224 ±5.25%	0.1508 ±0.14%	1.7854 ±0.28%	6.3535 ±1.89%
	300	0.1473 ±0.47%	1.3916 ±1.24%	4.8483 ±4.01%	0.1413 ±0.12%	1.4118 ±0.35%	4.2141 ±1.43%
	348	0.1458 ±0.32%	1.3409 ±0.85%	4.6141 ±3.69%	0.1398 ±0.10%	1.3281 ±0.33%	3.9661 ±1.44%
	360	0.1469 ±0.34%	1.2868 ±0.90%	4.9271 ±2.73%	0.1400 ±0.16%	1.2888 ±0.43%	4.0126 ±1.79%
	400	0.1440 ±0.34%	1.3591 ±1.05%	4.0064 ±3.48%	0.1382 ±0.25%	1.2954 ±0.67%	3.7257 ±2.14%
	500	0.1433 ±0.29%	1.3382 ±0.85%	3.9952 ±2.70%	0.1352 ±0.14%	1.3472 ±0.42%	3.1780 ±1.68%
	600	0.1416 ±0.33%	1.3760 ±0.94%	3.7782 ±3.14%	0.1359 ±0.13%	1.3399 ±0.38%	3.3536 ±1.49%
		HMM n=10			HMM n=11		
L_Q	L_S	λ	$10^4\lambda_2$	10^3K	λ	$10^4\lambda_2$	10^3K
348	150	0.1552 ±0.14%	2.2225 ±0.30%	6.7936 ±2.08%	0.1455 ±0.14%	2.3813 ±0.15%	4.9660 ±3.82%
	200	0.1459 ±0.22%	1.8336 ±0.37%	5.7585 ±3.30%	0.1417 ±0.17%	1.8428 ±0.35%	5.1264 ±2.07%
	300	0.1370 ±0.22%	1.4024 ±0.56%	3.8087 ±1.79%	0.1324 ±0.27%	1.3842 ±0.68%	3.2129 ±2.79%
	348	0.1353 ±0.15%	1.2962 ±0.38%	3.5507 ±1.68%	0.1316 ±0.22%	1.2518 ±0.69%	3.1546 ±1.94%
	360	0.1343 ±0.13%	1.2830 ±0.36%	3.4674 ±1.39%	0.1297 ±0.25%	1.2737 ±0.52%	2.9445 ±2.81%
	400	0.1334 ±0.16%	1.2602 ±0.38%	3.2164 ±1.71%	0.1302 ±0.20%	1.2160 ±0.56%	2.9704 ±1.59%
	500	0.1307 ±0.16%	1.3013 ±0.46%	2.8331 ±1.22%	0.1280 ±0.30%	1.2426 ±0.86%	2.7433 ±2.73%
	600	0.1305 ±0.23%	1.3097 ±0.56%	2.8239 ±1.82%	0.1257 ±0.22%	1.2908 ±0.55%	2.4921 ±1.79%

The table shows the fit parameters of the score distribution Prob($S = s|$ # of helices = n) for $1 \leq n \leq 11$ for $L_Q = 348$ and different subject lengths. For entries, where λ_2 is left out, a suitable fit (with a small reduced χ^2 value) to the modified Gumbel distribution Eq. (10) was not possible and only the Gumbel parameters of the high probability region are shown.

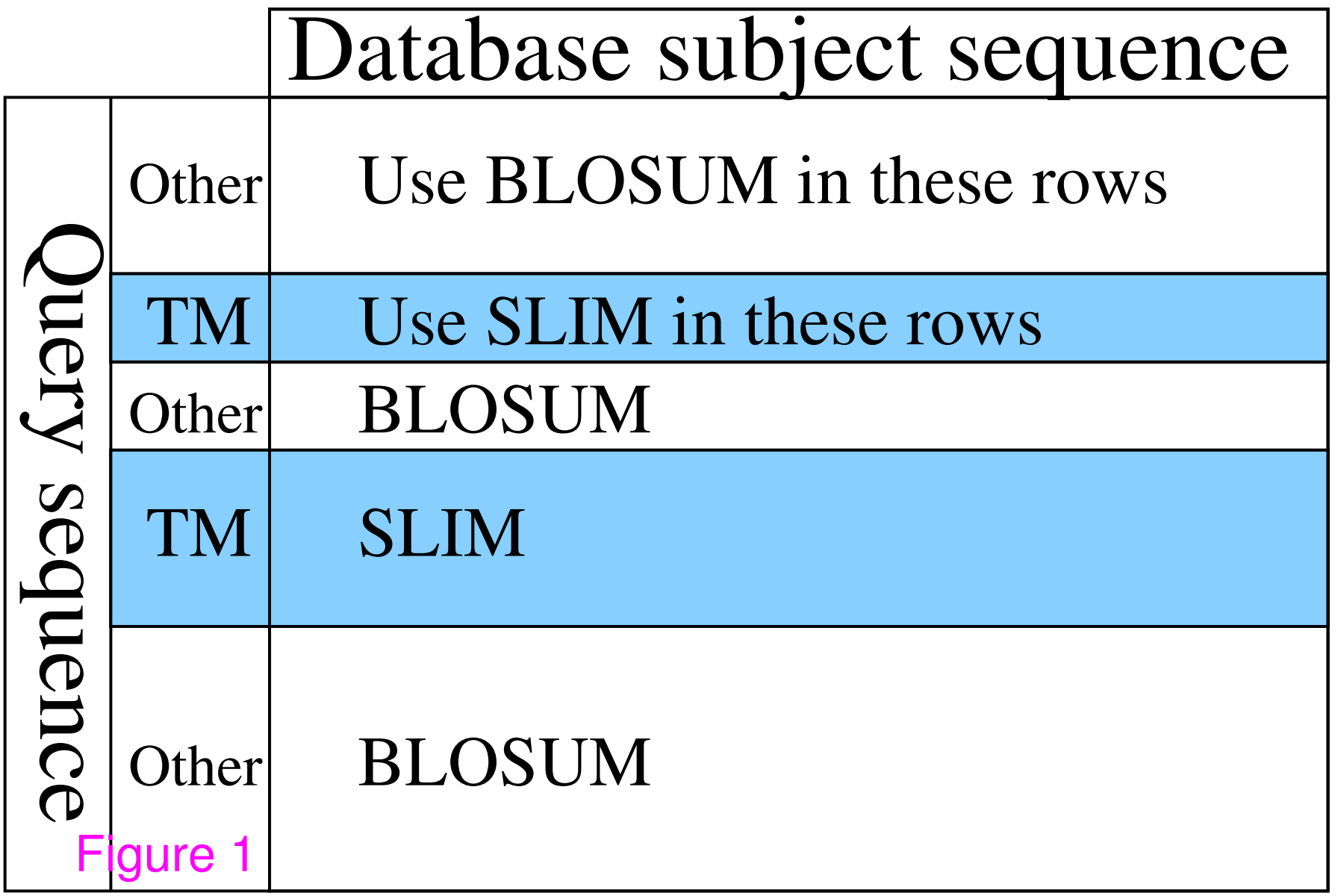

Database subject sequence
Query sequence
Other
Use BLOSUM in these rows
TM
Use SLIM in these rows
Other
BLOSUM
TM
SLIM
Other
BLOSUM

Figure 1

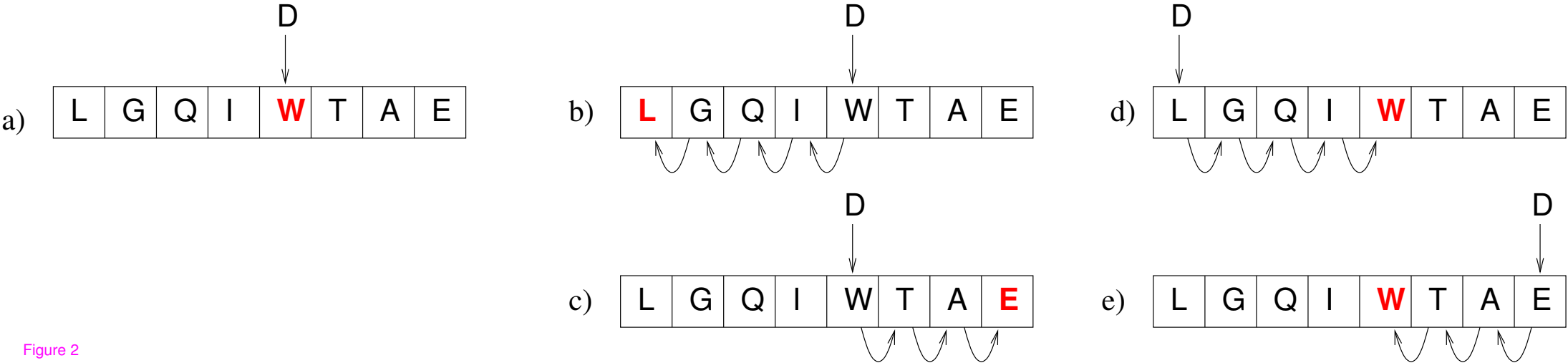

a)
D
L G Q I W T A E
b)
D
L G Q I W T A E
c)
D
L G Q I W T A E
d)
D
L G Q I W T A E
e)
D
L G Q I W T A E

Figure 2

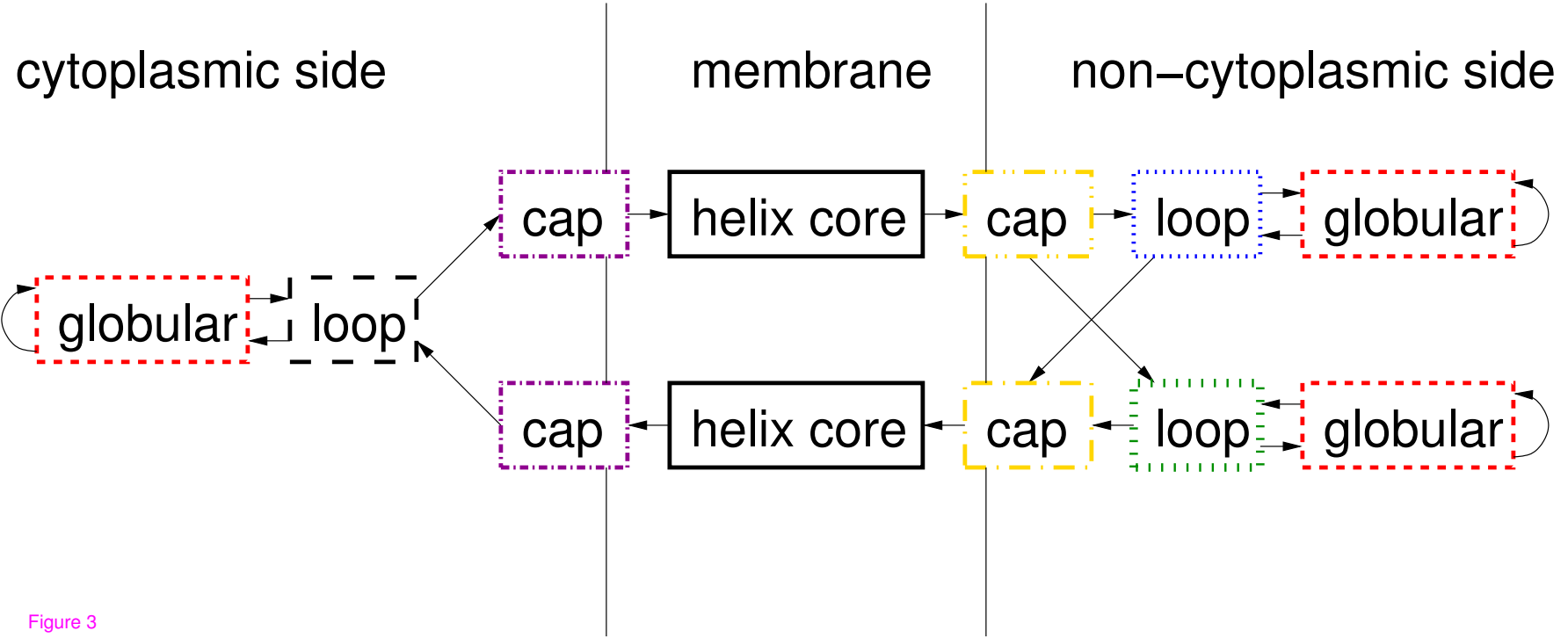
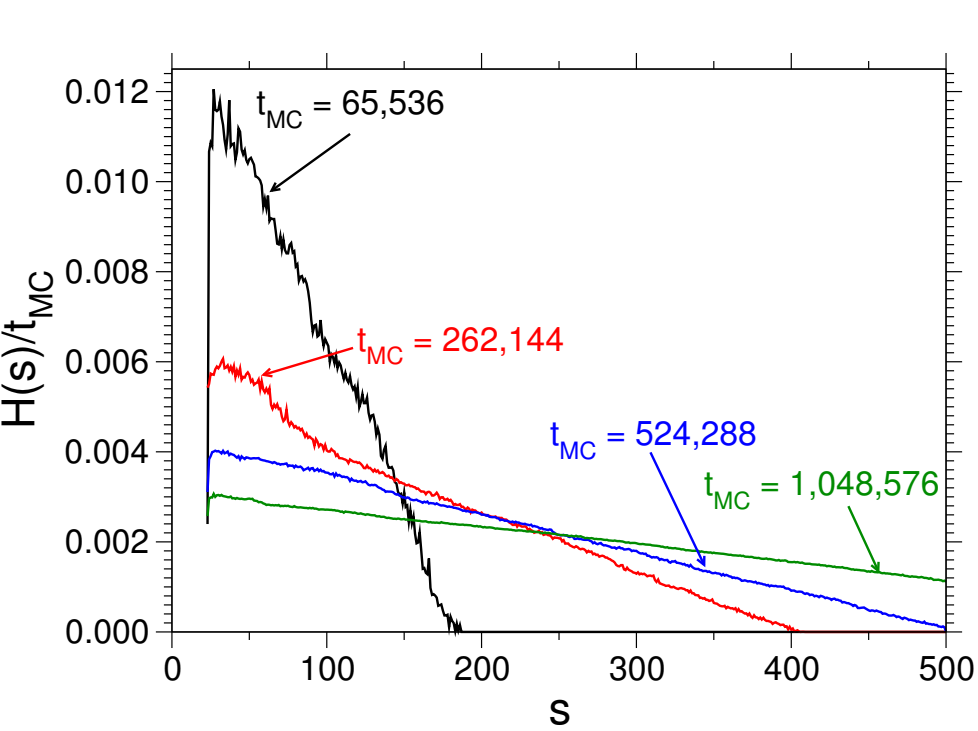
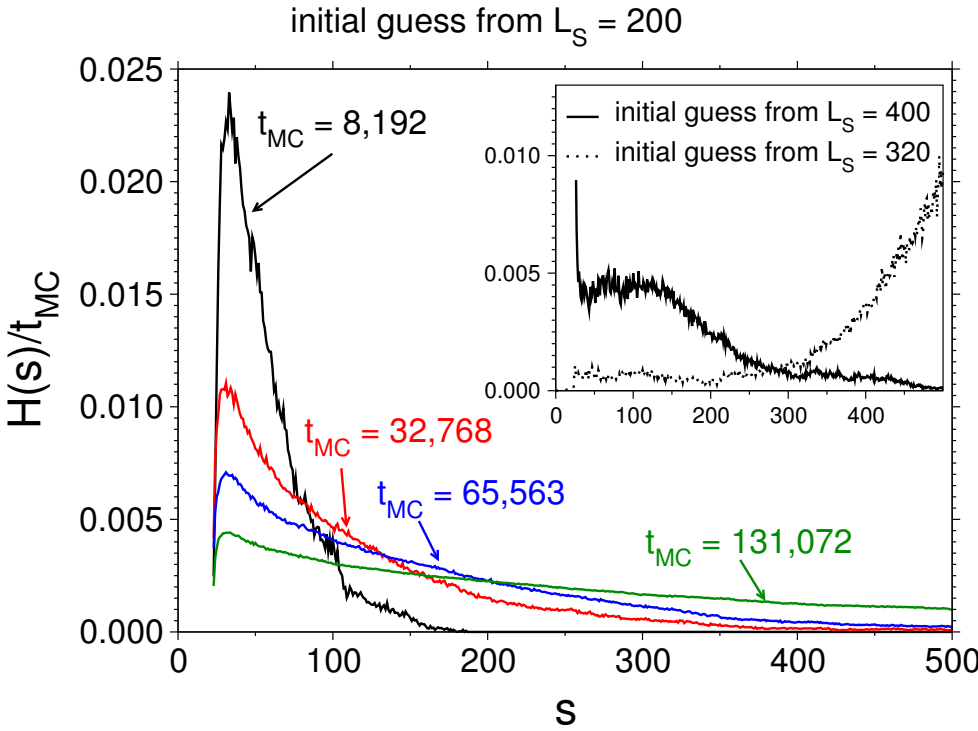

cytoplasmic side
membrane
non–cytoplasmic side
globular
loop
cap
helix core
cap
loop
globular
cap
helix core
cap
loop
globular

Figure 3

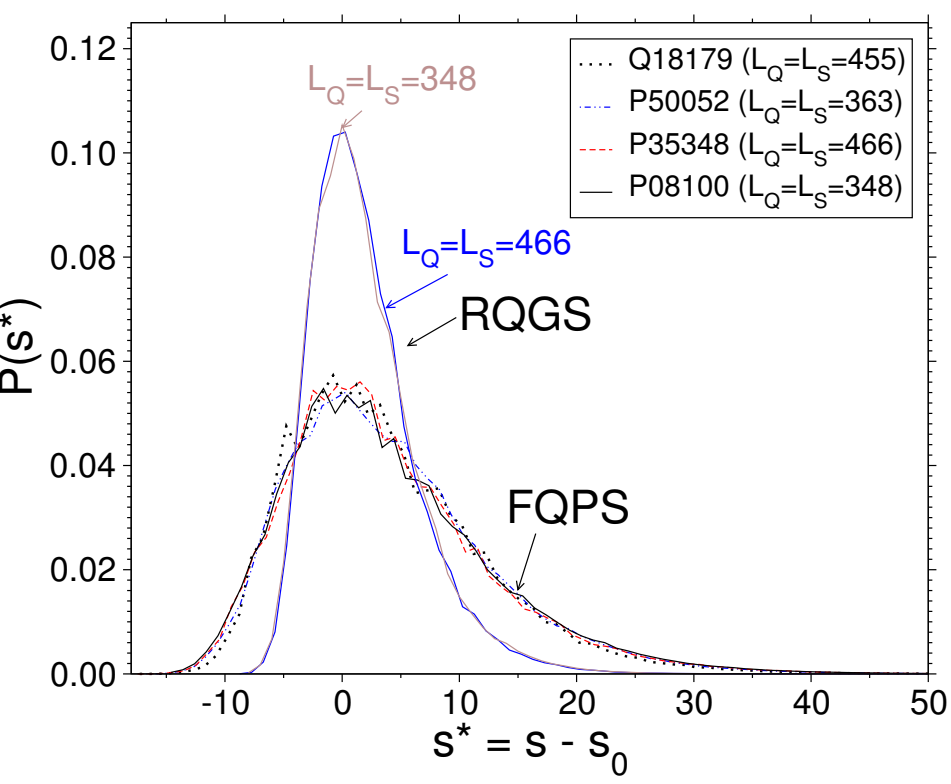

0.012
0.010
0.008
0.006
0.004
0.002
0.000
H(s)/t_MC
t_MC = 65,536
t_MC = 262,144
t_MC = 524,288
t_MC = 1,048,576
0
100
200
300
400
500
s

(a)

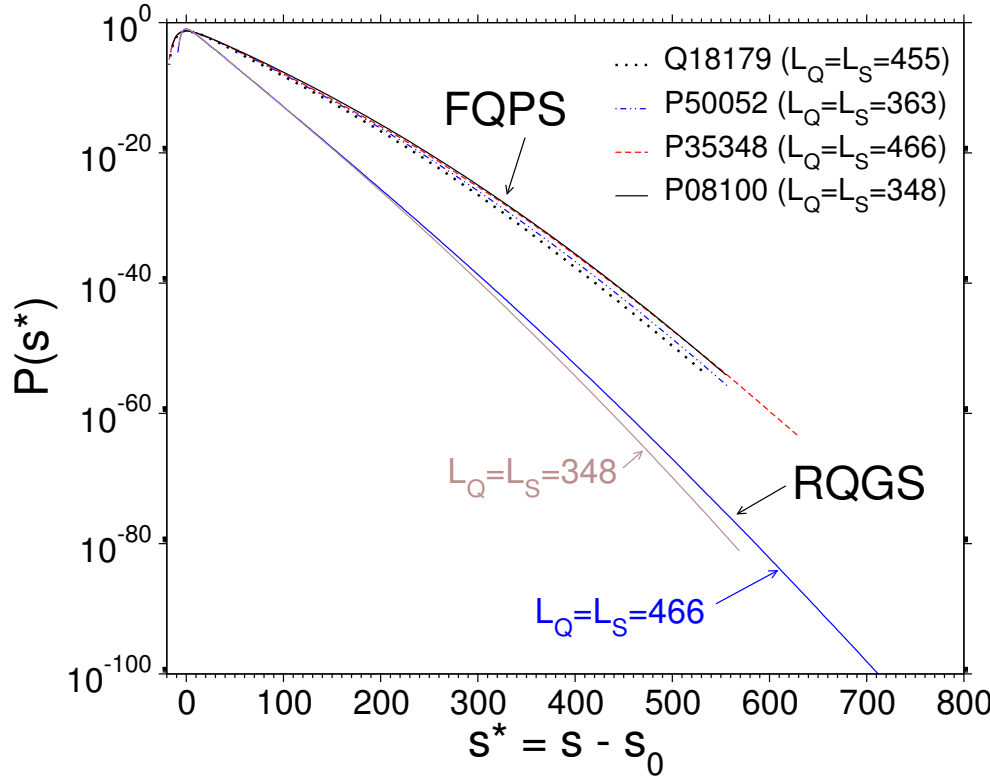

initial guess from L_S = 200
0.025
0.020
0.015
0.010
0.005
0.000
H(s)/t_MC
t_MC = 8,192
t_MC = 32,768
t_MC = 65,563
t_MC = 131,072
initial guess from L_S = 400
initial guess from L_S = 320
0.010
0.005
0.000
0
100
200
300
400
0
100
200
300
400
500
s

(b)

Figure 4


Q18179 (L_Q=L_S=455)
P50052 (L_Q=L_S=363)
P35348 (L_Q=L_S=466)
P08100 (L_Q=L_S=348)
L_Q=L_S=348
L_Q=L_S=466
RQGS
FQPS
P(s*)
s* = s - s_0
0.12
0.10
0.08
0.06
0.04
0.02
0.00
-10
0
10
20
30
40
50

(a)


Q18179 (L_Q=L_S=455)
P50052 (L_Q=L_S=363)
P35348 (L_Q=L_S=466)
P08100 (L_Q=L_S=348)
FQPS
L_Q=L_S=348
L_Q=L_S=466
RQGS
P(s*)
s* = s - s_0
10^{0}
10^{-20}
10^{-40}
10^{-60}
10^{-80}
10^{-100}
0
100
200
300
400
500
600
700
800

(b)

Figure 5

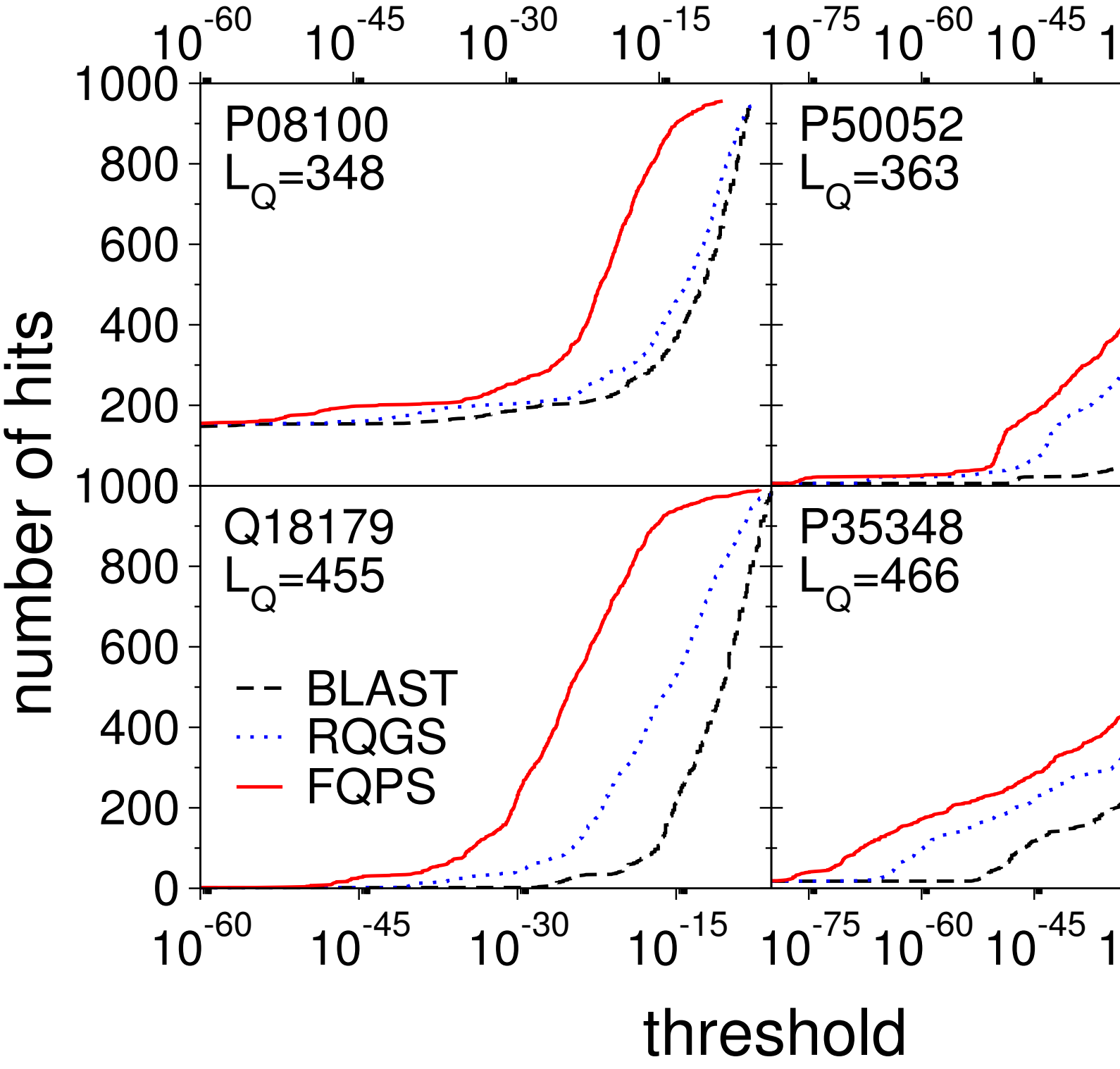
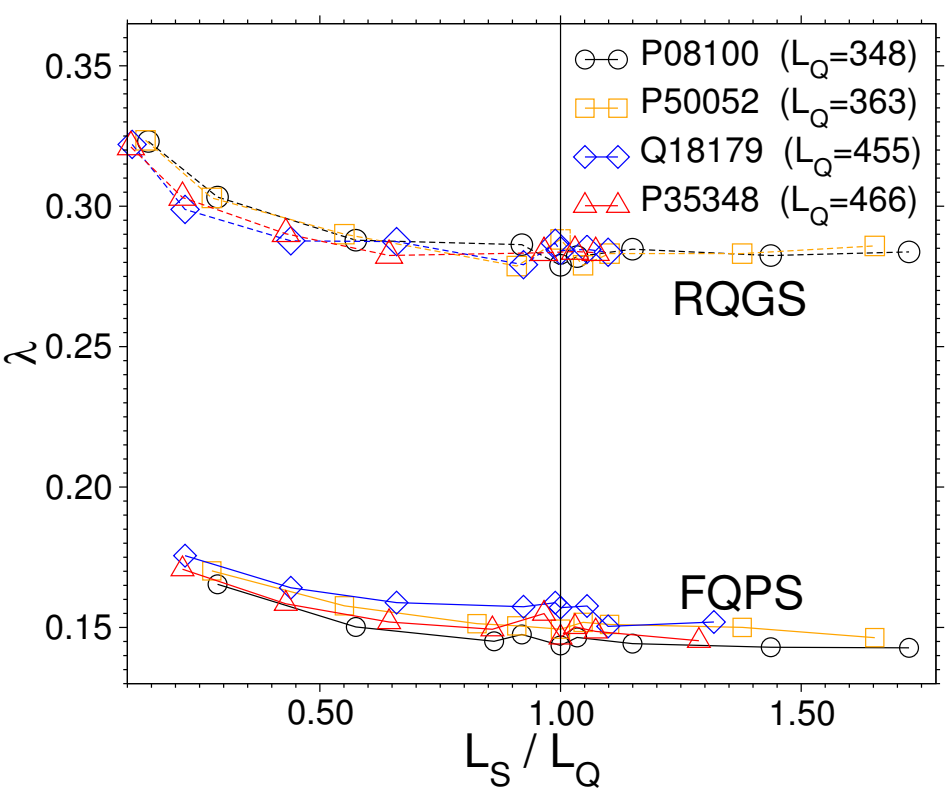
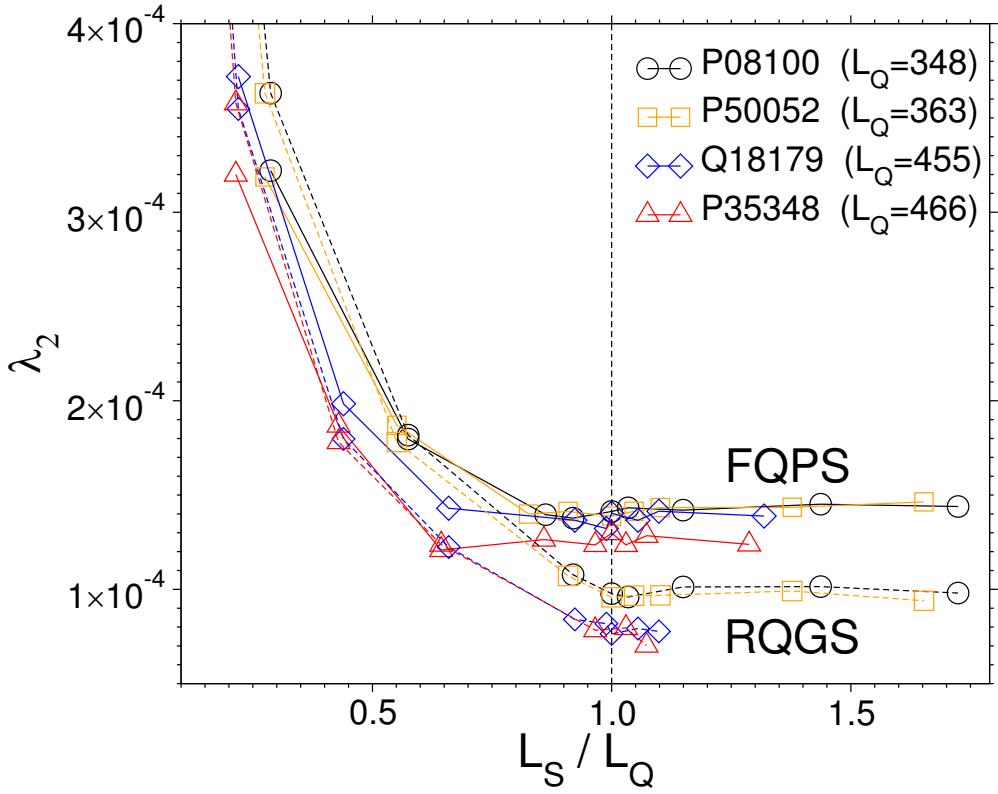

P08100
L_Q=348
P50052
L_Q=363
Q18179
L_Q=455
P35348
L_Q=466
BLAST
RQGS
FQPS
number of hits
threshold
0
200
400
600
800
1000
10^-60
10^-45
10^-30
10^-15
10^-75
10^-60
10^-45

Figure 6


P08100 (L_Q=348)
P50052 (L_Q=363)
Q18179 (L_Q=455)
P35348 (L_Q=466)
RQGS
FQPS
λ
0.35
0.30
0.25
0.20
0.15
0.50
1.00
1.50
L_S / L_Q

(a)


P08100 (L_Q=348)
P50052 (L_Q=363)
Q18179 (L_Q=455)
P35348 (L_Q=466)
FQPS
RQGS
λ_2
4×10^-4
3×10^-4
2×10^-4
1×10^-4
0.5
1.0
1.5
L_S / L_Q

(b)

Figure 7

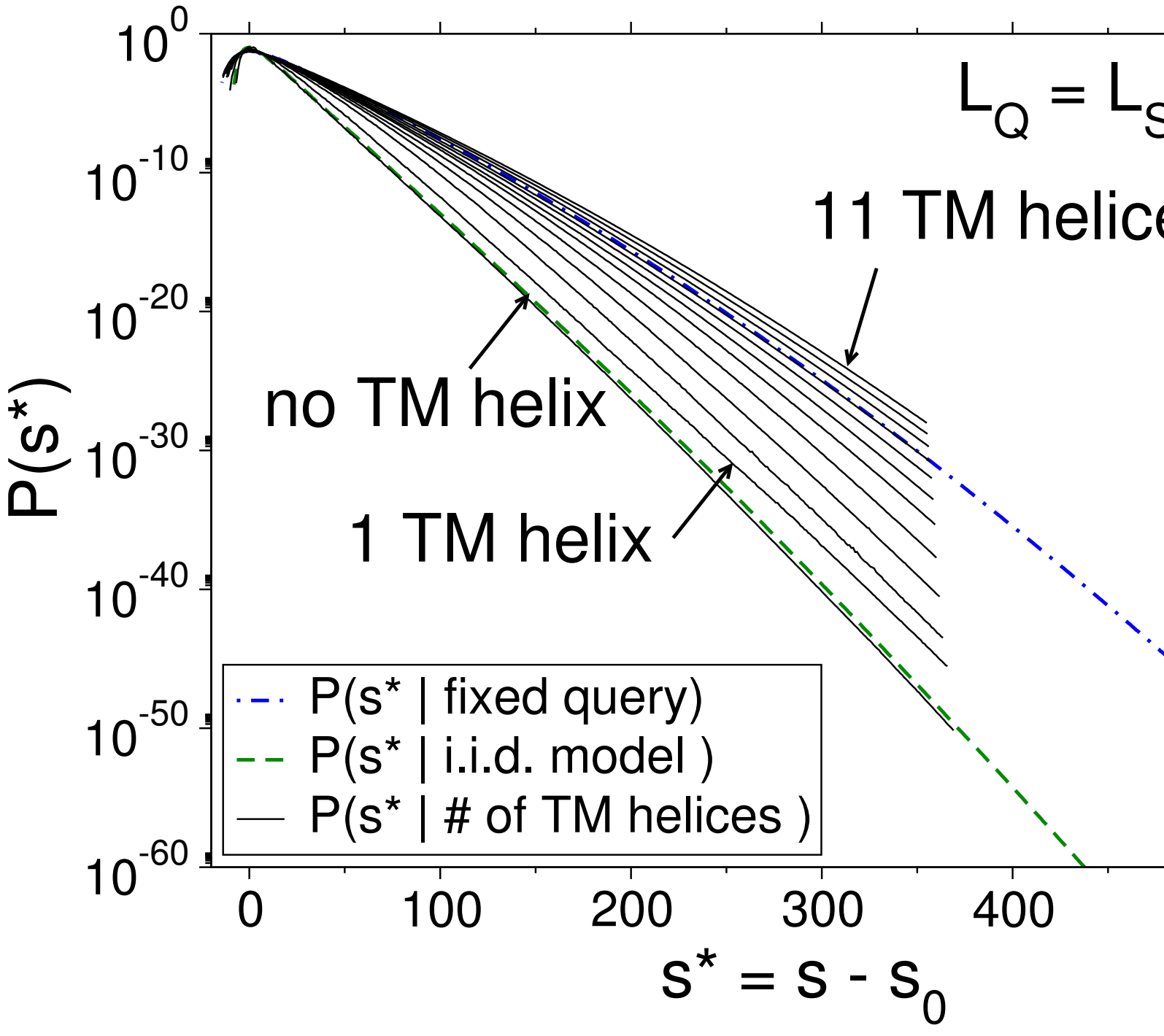

10^{0}
10^{-10}
10^{-20}
10^{-30}
10^{-40}
10^{-50}
10^{-60}
P(s*)
0
100
200
300
400
s* = s - s0
LQ = Ls
11 TM helice
no TM helix
1 TM helix
P(s* | fixed query)
P(s* | i.i.d. model)
P(s* | # of TM helices)

Figure 8

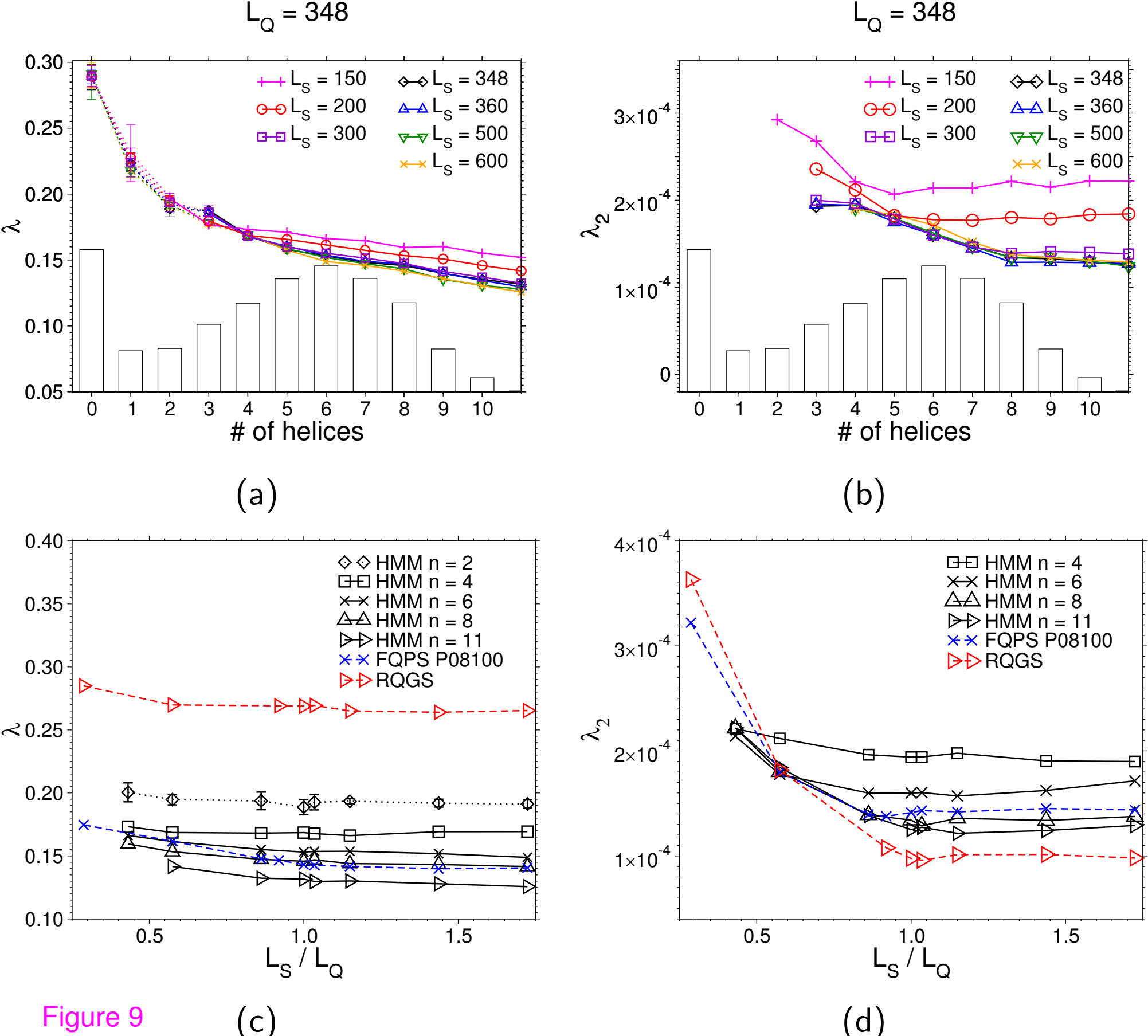

L_Q = 348
L_S = 150
L_S = 200
L_S = 300
L_S = 348
L_S = 360
L_S = 500
L_S = 600
λ
# of helices
(a)
L_Q = 348
λ_2
# of helices
(b)
HMM n = 2
HMM n = 4
HMM n = 6
HMM n = 8
HMM n = 11
FQPS P08100
RQGS
λ
L_S / L_Q
(c)
λ_2
L_S / L_Q
(d)

Figure 9