Submitted for publication.

Estimating and evaluating the statistics of gapped local-alignment scores

Timothy L. Bailey* and Michael Gribskov
tbailey@sdsc.edu gribskov@sdsc.edu

Abstract

We present a novel maximum likelihood-based algorithm for estimating the distribution of alignment scores from the scores of unrelated sequences in a database search. Using a new method for measuring the accuracy of p-values, we show that our maximum likelihood-based algorithm is more accurate than existing regression-based and lookup table methods. We explore a more sophisticated way of modeling and estimating the score distributions (using a two component mixture model and expectation maximization), but conclude that this does not improve significantly over simply ignoring scores with small E-values during estimation. Finally, we measure the classification accuracy of p-values estimated in different ways, and observe that inaccurate p-values can, somewhat paradoxically, lead to higher classification accuracy. We explain this paradox and argue that statistical accuracy, not classification accuracy, should be the primary criterion in comparisons of similarity search methods that return p-values that adjust for target sequence length.

Introduction

Sequence similarity searches are probably the single most widely used bioinformatics tool. The similarity between the query and a sequence in the database being searched is often defined in terms of the "gapped, local-alignment score". The alignment score for each possible pair of symbols in the sequences is specified by a scoring table. Due to insertions and deletions in sequences over the course of evolution, it is usually useful to allow gaps to be inserted into the sequences being aligned. An alignment cost is specified for gaps of different lengths. The gapped, local-alignment score is computed by aligning a region of the query with a region of the target sequence, with gaps in either sequence inserted as necessary, to maximize the the sum of the scores for pairs of aligned symbols plus the cost for gaps.

It is well recognized that accurate measures of the statistical significance of the alignment scores greatly enhance the usefulness of similarity searches. Knowing the distribution of the alignment scores of unrelated sequences allows, for example, the expected number of false positives at a given threshold to be estimated. This paper is concerned with the problem of estimating the statistical significance of such gapped local-alignment scores. It also addresses the problem of measuring and comparing the accuracy of these significance estimates.

Although a usable theory for the distribution of *ungapped*, local-alignment scores exists (Karlin and Altschul 1990), no such theory exists for gapped local-alignment scores. Currently, most sequence similarity algorithms estimate gapped alignment score significance in one of two basic ways. One type of method is a lookup table approach. It assumes a fixed random sequence model and a parametric distribution function and precalculates the parameters of the distribution for a variety of scoring table/gap penalty combinations (Altschul and Gish 1996; Altschul *et al.* 1997). In the lookup table ap-

San Diego Supercomputer Center, 9500 Gilman Drive, La Jolla, California 92093-0537

*To whom correspondence should be addressed.

proach, the estimation of the parameters generally utilizes simulated sequences. This approach is used by, for example, the BLAST algorithm (Altschul *et al.* 1997).

The other type of method in current use is sometimes referred to as "empirical", because it estimates the parameters of the distribution function directly from the scores observed in the database search. It is similar to the lookup table approach in that it assumes a parametric distribution function, but differs in that the parameters are reestimated for each database search rather than being precalculated. The empirical approach precludes the need to assume a particular random sequence model, and allows the use of a wider variety of scoring tables and gap penalties. This type of approach is used by, for example, the FASTA algorithm (Pearson 1998) and HMMER (Eddy 1995). Recently, a heuristic approximation of the gapped local-alignment score distribution was described (Mott and Tribe 1999).

This article describes an investigation of empirical methods of score distribution estimation. We chose to study empirical methods due to their several advantages over lookup table and heuristic approximation methods. As mentioned above, lookup table methods restrict the user of the search algorithm to the predefined combinations of gap penalties and scoring tables for which parameters have been estimated. Empirical methods do not suffer from this restriction. The other advantages enjoyed by empirical methods over lookup table methods arise from the fact that their parameter estimates are directly influenced by the length and composition of both the query and the target sequences in the database being searched. The fixed model for random sequences assumed by lookup table methods is often not an accurate representation of the actual sequences in the database being searched. This lack of fidelity in the assumed sequence model can have a large affect on the estimates of score significance. Since empirical methods use the actual observed scores, the distribution parameters they estimate automatically reflect the composition of the query and database sequences. Empirical methods also can compensate somewhat for what are known as "edge effects", a problem inherent in all current methods (including the heuristic approximation method). Edge effects cause the parametric distribution functions not to hold for short sequences, and are due to the fact that the distribution functions do not account well for the ends (edges) of the two sequences being aligned. As pointed out in Mott and Tribe (1999), empirical methods implicitly adjust the parameter estimates to account for edge effects, whereas the heuristic approximation method is unsafe when the sequence lengths are short or the lengths are very dissimilar. The heuristic approximation method is also unreliable when the expected length of local alignments is a significant fraction of the sequences' lengths, which can easily occur with short sequences, low entropy scoring tables (e.g., high PAM-number PAM tables), or low gap penalties. Once again, empirical methods are known to be able to ameliorate these problems (Mott 1992; Spang and Vingron 1998)

On the other hand, empirical methods for estimating gapped local-alignment score distributions have limitations relative to lookup and heuristic approximation methods. Empirical methods must somehow distinguish between the scores of related and unrelated target sequences in order to estimate the distribution parameters from just the unrelated sequence scores. Other limitations of empirical methods are due to the requirement that the database being searched contain a "sufficient" number of sequences unrelated to the query for accurate estimates of the distribution parameters to be found. Clearly, empirical methods should not be used if the database being searched is very small or contains few sequences that are dissimilar to the query and to each other. Most biological se-

quence databases contain thousands of diverse sequences, so this limitation is usually not encountered in practice.

In this paper, we present a new empirical estimation procedure that minimizes the limitations mentioned in the preceding paragraph. Our algorithm is based on maximum likelihood estimation, and it improves on existing empirical approaches in several ways. Firstly, compared with with existing regression-based empirical algorithms, we show that maximum likelihood gives better estimates of score significance, especially with datasets containing relatively few sequences. This reduces the limitation concerning the number of unrelated sequences required by empirical methods. Secondly, our algorithm improves upon the maximum likelihood based method used in the /hmmer/ system (unpublished manuscript, `ftp.genetics.wustl.edu/pub/eddy/papers/evd.ps`), by directly addressing the edge effect mentioned above. An additional improvement on previous maximum likelihood methods is a technique for stratifying the sequences in the database by length and estimating score distributions for each of the length ranges separately. Score significance estimates for each target sequence are then based on the two nearest length ranges. We show that this gives improved significance estimates compared to lookup table methods as well as existing empirical methods. We also investigate two ways of handling the other limitation of empirical methods, the removal of scores of sequences related to the query prior to parameter estimation. We demonstrate that removing scores with estimated E-values less than one works as well as modeling the scores with a two-component mixture model. An additional contribution of this paper is a novel way of visualizing and measuring the accuracy of significance estimates that we use to comparatively evaluate our method and several existing methods.

Estimating the statistics of gapped local-alignment scores

Extreme value distribution

In this work, we assume that the scores of gapped local-alignments follow an extreme value distribution (Gumbel 1958; Lawless 1982; Santner and Duffy 1989) of the form

$$P(S > x) = 1 - \exp(-KNe^{-\lambda x}), \qquad (1)$$

where S is the score, N is the size of the "search space" (described below), and K and λ are parameters that depend on the compositions of the sequences and on the scoring system (score matrix plus gap penalties). Experimental evidence has shown that the scores from gapped local-alignments approximately follow this distribution (Altschul and Gish 1996) as long as the gap penalties are severe enough. This distribution has been used both in previous empirical methods (Pearson 1998) as well as lookup methods (Altschul *et al.* 1997). It has been shown that *ungapped* local-alignment scores asymptotically approach this distribution as the length of the two sequences and the scores go to infinity. The optimum gapped alignment is also the optimum ungapped alignment if the gap penalties are infinite. As the gap penalties become less severe, gapped alignments between unrelated sequences tend to get longer. Eventually, if the penalties are too weak, the expected alignment length approaches the lengths of the sequences, and the extreme value distribution becomes a very poor approximation.

The search space size, N, (Eqn. 1), bears a strong relation to why an extreme value distribution is appropriate to local alignment scores. The optimum alignment score between two sequences can be thought of as the score of the best alignment starting at some pair of positions, one in each sequence. If the query sequence has length q and the target sequence has length t, then the number of candidate starting positions for alignments is

$$N = qt.$$

In practice, this tends to overestimate the true search space size, since high-scoring alignments cannot start too near the edge of either sequence. Consequently, a commonly used heuristic is to subtract the expected length, l, of an alignment (of the query to an unrelated sequence of length t) from q and t, giving

$$N = (q - l)(t - l). \tag{2}$$

We follow Altschul and Gish (1996) and estimate l by

$$l = \log(Kqt)/H,$$

but we treat H as a parameter to be estimated. The reason for this is that we do not wish to assume a sequence model for either the query or the target sequence in order to compute H, since that would negate one of the advantages of the empirical approach we are pursuing.[1]

Equations 1 and 2 define the p-value of score x for a target sequence of length t. Simply, the p-value is the probability of an unrelated sequence of length t having the observed score, x, or greater. This definition of a p-value normalizes for the length and composition of both the query and target sequences. It is the same definition used by FASTA algorithm, but differs from that used by BLAST, which does not normalize p-values with respect to the lengths of the target sequences in which the alignment score is observed. We define the E-value of score x observed in a sequence of length t as the p-value, p, times the number of target sequences in the database searched, n:

$$E = np.$$

We use this definition in what follows in order to determine which sequences are related to the query.

[1] H is the expected score per pair of aligned symbols, whose value depends upon the scoring table and the sequence model(s), and can be computed analytically if these are known.

The MLH *algorithm*

We can now describe the maximum likelihood estimation procedure for empirically estimating the distribution of gapped local-alignment scores. We assume that the similarity score x_i for aligning a given query sequence to the ith *unrelated* target sequence has the probability density function

$$f(x_i) = \lambda K N_i \exp(-\lambda x_i - K N_i e^{-\lambda x_i}), \tag{3}$$

where $N_i = (q - l_i)(t_i - l_i)$ and $l_i = \log(Kqt_i)/H$, This is the probability density function corresponding to (one minus) the cumulative distribution function (Eqn. 1). The subscript i indicates that the score, x_i and search space size, N_i, depend on the particular target sequence in question. The value of N_i, in particular, depends on the length, t_i, of the target sequence.

We estimate the values of λ, K and H using maximum likelihood estimation given the scores for all the unrelated sequences in a database. (How we determine which sequences are unrelated will be addressed below.) We are given a list of scores, $X = (x_1, \ldots, x_n)$, the length of the query, q, and a list of target sequence lengths, $T = (t_i, \ldots, t_n)$, and we wish to find the values of λ and K that maximize the log likelihood function

$$\begin{aligned} L(K, \lambda, H) &= \sum_{i=1}^{n} \log f(x_i) \\ &= n \log(\lambda K) + \\ &\quad \sum_{i=1}^{n} (\log N_i - \lambda x_i - K N_i e^{-\lambda x_i}). \end{aligned} \tag{4}$$

The complete algorithm (MLH, outlined in Fig. 1) must determine which target sequences are not homologous to the query, since MLH is estimating the parameters for the unrelated sequence score distribution. The algorithm first performs maximum likelihood estimation using the scores from all target sequences. It does this by maximizing the log likelihood with respect

function MLH(X: *list of scores*, T: *list of lengths*, q: *length of query*)
Find the values of K, λ and H that maximize the likelihood function given the sequence scores and lengths.
Set initial $\hat{\lambda}$ to 1/(sample standard deviation of all scores), $\hat{H}$ to 1.
Mark all scores as "use".
for ($i = 0$; $i <$MAXITER; i++) **do**
Using scores marked "use", get maximum likelihood estimates for $\hat{K}$ and $\hat{\lambda}$, holding $\hat{H}$ constant, via the Newton-Raphson algorithm.
Using scores marked "use", get maximum likelihood estimate for $\hat{H}$, holding $\hat{K}$ and $\hat{\lambda}$ constant, via the Newton-Raphson algorithm.
if (improvement in log likelihood)/(log likelihood) $< 10^{-6}$ **then**
return $\hat{K}$, $\hat{\lambda}$ and $\hat{H}$.
end
Compute E-values of all scores using current $\hat{K}$, $\hat{\lambda}$ and $\hat{H}$.
Mark all scores as "use" except those with E-value < 1.
end
return $\hat{K}$, $\hat{\lambda}$ and $\hat{H}$.
end

Figure 1: **The MLH algorithm.**

to K and λ, with H held constant, and then with respect to H, holding the other two variables constant. (This proved to be faster and more robust than using conjugate gradient to simultaneously optimize over all three variables.) Once the maximum likelihood parameter estimates have been obtained, "outlier" scores are removed. We use the heuristic that sequences with E-values less than one are probably related to the query. Hence, sequences where

$$E \;=\; n(1 - \exp(-\hat{K} N_i e^{-\hat{\lambda} x_i})) < 1.0$$

are removed from from the set of sequences. Maximum likelihood estimation is then repeated, followed by outlier removal again.[2] These two steps–maximum likelihood estimation followed by outlier detection and removal–are repeated until the fractional change in log likelihood is less than a 10^{-6}.

[2]Outlier "removal" is temporary; the set of sequences from which sequences are removed at each iteration is the *original* set of target sequences, not the partial set from the previous iteration.

We treat scores with E-values less than one as outliers since, on average, only one unrelated sequence will have such a high score. One could argue for setting this threshold lower in order to increase the odds of including all unrelated sequence scores in the parameter estimation. On the other hand, this would increase the risk of including the scores of *related sequences*, which would severely affect the accuracy of the distribution estimate. Similarly, one could consider increasing the threshold to reduce this risk, and this would require treating the remaining scores as a censored or incomplete sample. As we describe in Results, we tested a more principled approach–using a two-component mixture to model the data–and found that using the simpler (and computationally faster) E-value threshold of one gave nearly identical results.

To optimize the log-likelihood function (Eqn. 4) with with respect to K and

λ, we require its partial derivatives,

$$\frac{\partial L}{\partial K} = \frac{n}{K} - \sum_{i=1}^{n} N_i e^{-\lambda x_i}, \text{ and} \quad (5)$$

$$\frac{\partial L}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} K N_i x_i e^{-\lambda x_i}. \quad (6)$$

To solve for the maximizing values $\hat{K}$ and $\hat{\lambda}$ we set Eqn. 6 to zero and solve for $\hat{K}$ giving

$$\hat{K} = \frac{n}{\sum_{i=1}^{n} N_i e^{-\hat{\lambda} x_i}}. \quad (7)$$

We then multiply Eqn. 6 by n (which does not affect the location of the maximum with respect to λ) and substitute $\hat{K}$ for K to get

$$g(\hat{\lambda}) = \frac{1}{\hat{\lambda}} - \frac{\sum_{i=1}^{n} x_i}{n} + \frac{\sum_{i=1}^{n} N_i x_i e^{-\hat{\lambda} x_i}}{\sum_{i=1}^{n} N_i e^{-\hat{\lambda} x_i}}. \quad (8)$$

We find the root of Eqn. 8 using the Newton-Raphson root-finding algorithm. This requires the first derivative of $g()$ with respect to $\hat{\lambda}$,

$$g'(\hat{\lambda}) = -\hat{\lambda}^{-2} - \frac{\sum_{i=1}^{n} N_i x_i^2 e^{-\hat{\lambda} x_i}}{\sum_{i=1}^{n} N_i e^{-\hat{\lambda} x_i}} + \left(\frac{\sum_{i=1}^{n} N_i x_i e^{-\hat{\lambda} x_i}}{\sum_{i=1}^{n} N_i e^{-\hat{\lambda} x_i}} \right)^2.$$

The Newton-Raphson algorithm starts with an initial estimate for $\hat{\lambda}$. We use one over the standard deviation of the scores. The algorithm then repeatedly computes $g(\hat{\lambda})$ until it is within some ϵ of zero (we use $\epsilon = 10^{-4}$). On each iteration, the new estimate is $\hat{\lambda} = \hat{\lambda} - \frac{g(\hat{\lambda})}{g'(\hat{\lambda})}$. The final value of $\hat{K}$ is obtained by substituting the final value of $\hat{\lambda}$ into Eqn. 7. Our experience shows that the smoothness of the function $g(\hat{\lambda})$ near the root is sufficient to preclude the need for more sophisticated root-finding methods (data not shown).

The second optimization, over H, also uses a modified version of the Newton-Raphson algorithm. It requires the first two partial derivatives of the log likelihood with respect to H. To compute these, it is convenient to define three quantities for each target sequence,

$$a_i = 2l_i - q - t_i,$$

$$b_i = \frac{1}{N_i} - K e^{-\lambda x_i}, \text{ and}$$

$$c_i = -\frac{l_i}{H}.$$

Holding K and λ constant, we can write the first two partial derivatives of the log likelihood with respect to H as

$$\frac{\partial L}{\partial H} = \sum_{i=1}^{n} a_i b_i c_i$$

$$\frac{\partial^2 L}{\partial H^2} = \sum_{i=1}^{n} \left(2 b_i c_i^2 - \left(\frac{a_i c_i}{N_i} \right)^2 - \frac{2 a_i b_i c_i}{H} \right).$$

As before, it is convenient to define the function $g()$ so that

$$g(\hat{H}) = \left. \frac{\partial L}{\partial H} \right|_{\hat{H}}, \text{ and}$$

$$g'(\hat{H}) = \left. \frac{\partial^2 L}{\partial H^2} \right|_{\hat{H}}.$$

Our modified Newton-Raphson search for determining $\hat{H}$, the value of H that minimizes the likelihood function for given values of K and λ, is shown in Fig. 2. It uses binary search until the second partial derivative is positive, then Newton-Raphson search. An additional binary search is done if $\hat{H}$ ever becomes negative since such values are meaningless. Iteration stops when the the absolute value of the first derivative becomes less than 10^{-4}.

Target length stratification: MLHS

Eqn. 1 is only asymptotically accurate in increasing target and query sequence length, even for ungapped alignment scores. For gapped alignment scores, it is a convenient parametric form with some empirical justification, but tends to be inaccurate for short sequences. As we

```
function MNR(K, λ, Ĥ, X, T, q)
  Find the value of H that maximizes the likelihood function
  for (i = 0; i <MAXITER; i++) do
    old_Ĥ = Ĥ
    Compute g(Ĥ) and g'(Ĥ)
    if |g(Ĥ)| < 10^-4 then return Ĥ end
    if g'(Ĥ) > 0 then
      Binary search in direction of first derivative:
      if g(Ĥ) > 0 then Ĥ = 2Ĥ else Ĥ = Ĥ/2 end
    else
      Newton-Raphson step:
      Ĥ = Ĥ − g(Ĥ)/g'(Ĥ)
      Do binary search if H is negative:
      if g(Ĥ) <= 0 then Ĥ = old_Ĥ/2 end
    end
  end
  return Ĥ
end
```

Figure 2: Modified Newton-Raphson algorithm to find optimal H.

show in Results, its accuracy is improved somewhat by the use of the edge effect correction (Eqn. 2). For databases containing more than 10,000 sequences unrelated to the query, we can further refine the accuracy of empirical p-value estimates by stratifying the scores for target sequences by their lengths, and estimating different values for the extreme value distribution parameters for each length group. In other words, we group the target sequences into two or more length ranges and execute algorithm MLH separately on the sequences in each length range. This yields a set of estimated extreme value distributions, each with improved accuracy for target sequences whose lengths fall in a particular range.

To compute a p-value for a target sequence of length t using such a set of extreme value distributions, we take the weighted geometric mean of its p-value according to the parameters for the two nearest length ranges. (Doing this smooths the p-values near the transitions between length ranges.) We calculate p_1, the p-value according to the parameters calculated for the length range containing t, and p_2, the p-value according to the other (bracketing) range's parameters. The weight, w, for p_1 varies from 0.5 if t is at the extreme edge of its range and 1.0 if t is at the midpoint of its range. Thus,

$$w = \begin{cases} 0.5 + 0.5(u-t)/(u-m), & \text{if } t \geq m, \\ 0.5 + 0.5(t-l)/(m-l), & \text{otherwise,} \end{cases}$$

where l, m and u are the minimum, midpoint and maximum, respectively, of the length range containing t. We than calculate the (smoothed) p-value as

$$p = p_1^w p_2^{1-w}.$$

(If t is below the lowest range's midpoint or above the highest range's midpoint, we use instead $p = p_1$.)

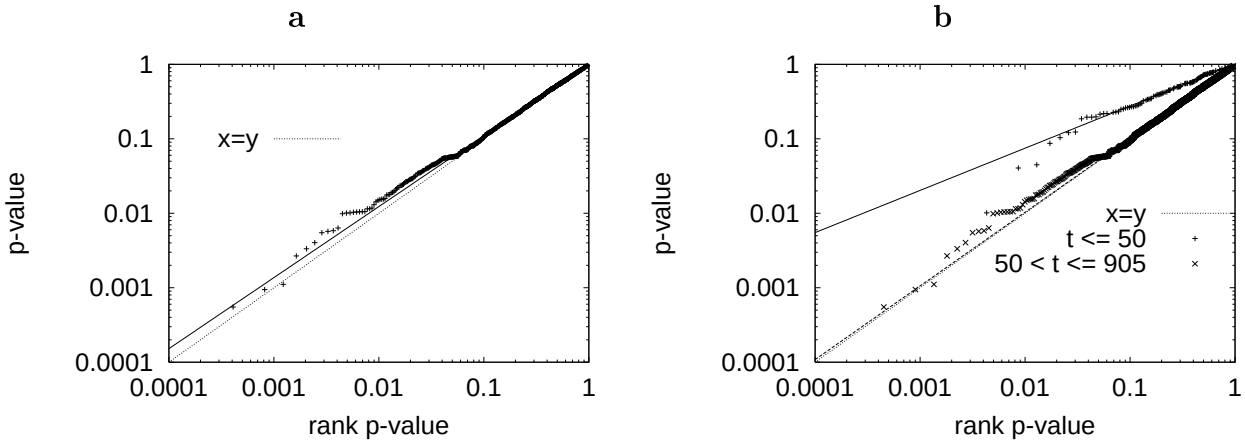

Figure 3: **Illustrating p-value slope error.** The plots illustrate the results of searching a database with a single query. Each point shows the p-value (p) versus the rank p-value (p_r) of the score of a single target sequence. Slope lines were estimated by weighted linear regression of $\log(p)$ versus $\log(p_r)$. The ideal slope is 1 and is indicated by the line $x = y$. Panel **a** shows the overall results for the database search, and panel **b** treats the sequences with lengths (t) less than or equal to fifty separately from the rest of the sequences.

Evaluating the statistics of alignment scores

Previous work had usually evaluated the quality of score distribution estimates using goodness-of-fit tests such as the Kolmogorov-Smirnov test (Pearson 1998) or the χ^2 test (Altschul *et al.* 1997). We introduce here a more descriptive metric, p-value slope error (PSE), that has the desirable quality of indicating both the magnitude and the predominant direction of the error in the p-values estimated by a particular method. The P-value slope error metric allows the accuracy of different score distribution methods to be compared, as well as giving a measure of the expected error in p-values of different sizes reported by a given method.

The idea behind the p-value slope error metric is illustrated in the two plots in Fig. 3. The two plots were both generated from the results of a single search of a database containing n sequences unrelated to the query. The points in the two plots are (p, p_r) where p is the rth smallest p-value reported for sequences unrelated to the query, and $p_r = r/(n+1)$. (We call p_r the "rank p-value".) These points should lie approximately along the line $x = y$ if the estimate of the score distribution is accurate. This is because the p-values of a discrete distribution should be approximately uniformly distributed. For example, if $n = 1$, there is only one unrelated sequence, and the expected value of the smallest (only) p is $p_r = r/(n+1) = 1/2$. Since p_r is the expected value of the rth smallest p-value, the points (p, p_r) should lie (approximately) on the line $x = y$.

When we plot (p, p_r) for all the unrelated sequences in the database (Fig. 3a), they do, indeed lie close to the ideal line. On the other hand, if we compute the ranks of the p-values separately for sequences less than and greater than fifty (Fig. 3b), the points for the shorter target sequences follow a different line in the log-log plot. For the sequences of fifty or fewer residues, the p-values (y-axis) are consistently larger than the rank p-values (x-axis), indicating that they are overestimates of the true p-values.

The p-value slope error metric is based on the empirical observation that the points (p, p_r), as

defined above, tend to follow a straight line in log-log plots such as those in Fig. 3, even when they do not lie along the ideal line, $x = y$. The lines shown in the figure (aside from the $x = y$ lines) are the weighted least-squares regression[3] lines of the points (p, p_r) to the linear equation

$$\log(p) \quad = \quad m \log(p_r) + b. \qquad (9)$$

We call these lines "p-value slope lines", and observe that their slopes give an indication of the direction and magnitude of the errors in the p-values. This is the motivation for the p-value slope error metric, defined as

$$PSE \quad = \quad 1 - m.$$

Here m is the p-value slope, as defined above in Eqn. 9. As defined, a positive value of PSE indicates that the estimated p-values are too large on average.

We use p-value slope error to compare the statistical accuracy of different methods of estimating the distributions of alignment score. For meaningful comparisons, we average the *absolute value* of PSE over a number of experiments. In addition, we sometimes compute the slope error separately for target sequences in different length ranges. This provides a better picture of the statistical accuracies of different estimation methods.

Results

In this section we evaluate the statistical accuracy of our maximum likelihood-based method and several published p-value estimation methods. We use scores from Smith-Waterman gapped local-alignment algorithm (Smith and Waterman 1981) searches of small- (SCOP (Brenner *et al.* 1998), 2448 domain sequences) and medium-sized (SWISSPROT (Bairoch and Apweiler 1994), 87272 sequences) databases. We use the BLOSUM62 scoring table and gap opening and extension penalties of 11 and 1, respectively in all searches. The queries in the searches of the SCOP domain sequence database are each of the sequences in that database.[4] For searching SWISSPROT, we use each of the (shuffled) sequences in the *Mycoplasma genitalium* genome. We shuffle the letters in the *M. gen.* queries to avoid the problem of deciding which sequences in SWISSPROT are truly unrelated to the query sequence when calculating the p-value slope error. This is not necessary with the SCOP queries because we use the SCOP annotation to determine relatedness.

The results of the statistical accuracy comparison are shown in Table 1. Each value in the table is computed by first averaging the p-value slope errors for target sequences in given length ranges, and then averaging the *absolute values* of those averages. We average over several length ranges and average the absolute value of the error in each length range so that the overall errors given in the table are smaller for estimation methods whose p-values are accurate for all target sequence lengths. We choose length ranges that contain roughly equal numbers of target sequences so that each sequence in the database contributes equally to the average error. Five length ranges are used for SCOP target sequences and ten ranges for the larger SWISSPROT database.

We draw several conclusions from the statistical accuracy results shown in Table 1. Firstly, maximum likelihood with edge effect correction (MLH) appears to better utilize the information available in the scores of unrelated sequences in a small database than the regression-based

[3]We use weighted regression because the smallest log p-values have higher variance and might otherwise unduly influence the regression. As the an estimate of standard error (weight) of the rth p-value, we use $\sqrt{1/r}$.

[4]The SCOP database groups proteins into a hierarchical classification: class, fold, superfamily, family, protein. Along with the hierarchical classification, SCOP provides the domain sequences for each classified protein. As test cases, we use each of the 321 SCOP superfamilies that contain 2 or more sequences in the SCOP Version 1.37 domain sequence file that has been purged to contain no pairs of sequences that are more than 40% identical (`ftp://scop.mrc-lmb.cam.ac.uk/pub/scop/pdbd/pdb40d_1.37`).

estimation method	*database searched*	
	SCOP	SWISSPROT
MLHS	0.012	0.030
MLH	0.012	0.045
ML	0.030	0.049
EM	0.029	*not tested*
REGRESS1	0.033	0.059
REGRESS2	0.027	0.045
REGRESS3	0.036	0.048
ALTSCHUL-GISH	0.092	0.100

Table 1: **Statistical accuracy of different p-value estimation methods.** The values shown are the average p-value slope errors, computed as described in the text. The methods are: ML, the maximum likelihood method; MLH, the maximum likelihood method with the edge effect correction; MLHS, same as MLH with length stratification; EM, same as ML modeling unrelated and related scores with a mixture model; REGRESS1, REGRESS2 and REGRESS3, regression-based estimation methods; ALTSCHUL-GISH, lookup table method.

methods described by Pearson (1998). (We review these regression-based methods in the Appendix.) We base this conclusion on the fact that the average slope error for the best regression method (REGRESS2) is more than twice that of method MLH on the SCOP database (2448 sequences). These two methods have identical average slope error (0.045) with the much larger, SWISSPROT database. Secondly, using a two-component mixture to model the scores of related and unrelated sequences (method EM) does not make a significant improvement in comparison to simply removing scores with E-values less than 1.0. This conclusion is due to the negligible difference in average slope error between methods ML and EM, which differ only in how they detect and remove related sequence scores. (We did not test method EM with the SWISSPROT database due to its failure to improve over method ML and its very long execution time.) Thirdly, using length stratification greatly improves p-value accuracy when the database contains many sequences unrelated to the query. The average slope error for the searches of the SWISSPROT database (87272 sequences) is only 0.30 using length stratification (MLHS), compared with 0.45 without length stratification (MLH).[5] Among all the p-value estimation methods, MLHS has lowest average slope error. Fourthly, the lookup table method (ALTSCHUL-GISH) yields p-values that are far less accurate on average than either the maximum likelihood-based or regression based methods. In the SCOP searches, the average slope error using the lookup table method was almost nine times higher than using method MLHS; in the SWISSPROT searches, the error was over three times higher for the lookup method.

It is illustrative to plot the p-value slope error of the different estimation methods as a function of target sequence length. The plots in Figure 4 show the average p-value slope error as a function of target sequence length calculated for the maximum likelihood methods, the best regression-based method (REGRESS2) and the lookup table method (ALTSCHUL-GISH). The same data as was used to create these plots as

[5]We use length ranges that contain approximately 10000 sequences each. With the SWISSPROT database, this results in four ranges with upper lengths of 94, 188, 376 and 6486, respectively. The SCOP database contains only 2448 sequences, so there is only one length range, and methods MLH and MLHS are identical in this case.

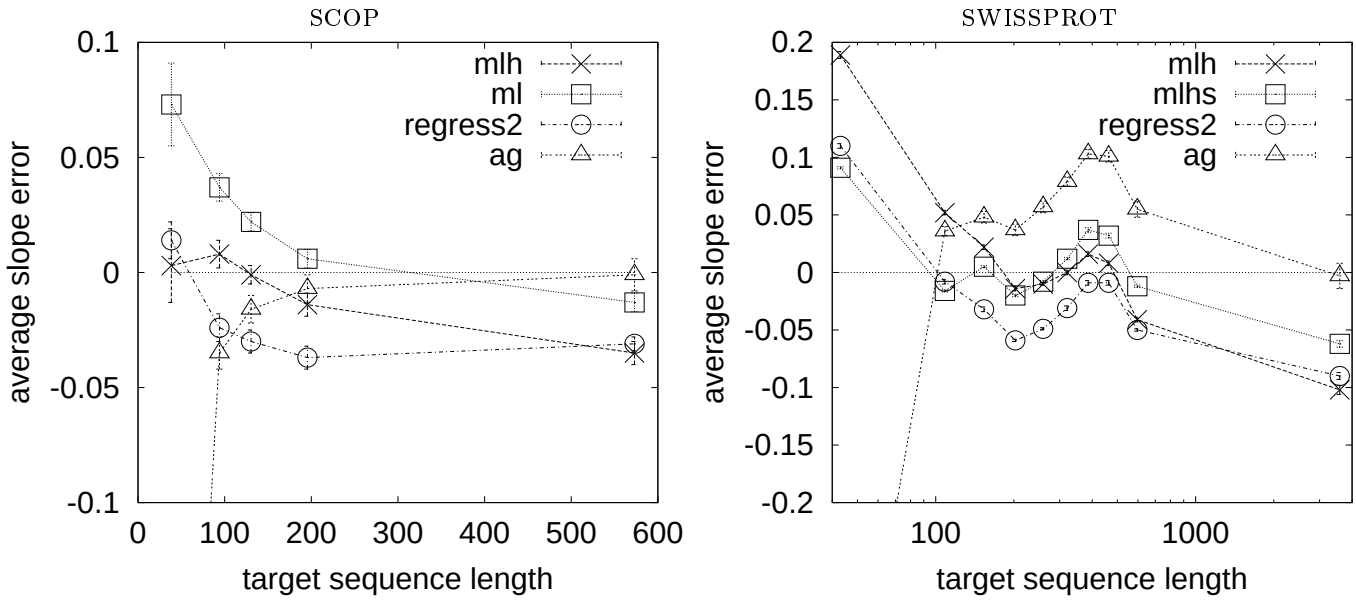

Figure 4: **Statistical accuracy as a function of target sequence length.** The plots show the average p-value slope error of the different p-value estimation methods as a function of target sequence length. Error bars show the standard error of the averages. Positive error indicates that the estimated p-values tend to be too large. The lookup table method (ALTSCHUL-GISH) is abbreviated "ag"; abbreviations for other methods are defined in the caption for Table 1.

was used to make Table 1. (The absolute values of the y-coordinates in the plots are averaged together for each method to give the values in Table 1.)

We can draw several additional conclusions from the two plots in Figure 4. Firstly, length stratification (MLHS) improves over the other methods by increasing the accuracy of p-values for both long and short target sequences. The curve for method MLHS is generally closer to zero, compared with all of the other methods in the right-hand plot. Without length stratification (method MLH), if the database is large (right-hand plot), method REGRESS1 is more accurate for sequences shorter than 100 residues, but generally less accurate for longer sequences. Secondly, both the regression-based and maximum likelihood-based methods tend to underestimate the significance (overestimate the p-value) of short target sequences, and to overestimate the significance (underestimate the p-value) of longer sequences. This is seen in the generally downward trend of the curves for methods REGRESS1, ML, MLH and MLHS. These methods, therefore, "favor" longer sequences to varying extents. This effect is least pronounced for the most accurate estimation method, MLHS. Thirdly, the lookup method, ALTSCHUL-GISH, greatly overestimates the significance (underestimates the p-value) of very short sequences, but is relatively accurate for long sequences. However, with the shuffled *M. gen.* queries and the SWISSPROT database, there is a tendency for the p-values to be too large for most target lengths.

Classification accuracy of estimated p-values

In this section we compare the ability of score p-values estimated by various methods to accurately classify protein sequences. We use the SCOP domain sequences mentioned earlier as the targets and queries. For deciding which sequences are related to each other, we adopt the

method	*RFP*	*Median RFP*	ROC_{50}	$E() < 0.02$	*E-number*
ML	5.1e-09	5.7e-10	–	**0.11**	–
EM	3.4e-09	5.3e-08	**0.85**	**0.27**	**0.25**
REGRESS1	**0.19**	**0.72**	**0.013**	**0.077**	**0.22**
REGRESS2	**0.15**	**0.9**	**0.018**	**0.092**	**0.22**
REGRESS3	**0.12**	**0.3**	**0.00052**	**0.0044**	**0.013**
ALTSCHUL-GISH	**5.4e-11**	**1.8e-12**	**4.1e-06**	**0.0025**	**0.021**

Table 2: **Classification accuracy relative to MLH.** The p-value of a two-tailed, *sign test* comparing method MLH to other methods. **Bold face** indicates that method MLH has superior accuracy on more test cases. Normal face indicates that the method named on the left has superior accuracy on as many or more test cases. A dash indicates that the two methods have superior accuracy on equal numbers of test cases.

same policy as used by Jaakkola *et al.* (1999): a target sequence is considered related to the query if it is in the same SCOP superfamily, unrelated if it is in a different SCOP fold, and ignored if it is in the same fold but a different superfamily. This policy is designed to ignore most target sequences that may have been misclassified in SCOP.

We use five evaluation methods for evaluating classification accuracy here: rate of false positives (RFP), median RFP, receiver operating characteristic 50 (ROC_{50}), the number of errors at an E-value threshold of 0.02 ($E() < 0.02$) and the E-number. Each takes the output of a single search and computes a quality score based on the order in which the positives (correct matches to the query) and negatives (non-matches to the query) appear in the output list. (These accuracy metrics are described in more detail below.) We then compare pairs of estimation methods by applying a two-tailed *sign test* to their classification accuracies (Chatfield 1983) for each of the queries.

The different classification accuracy evaluation methods emphasize distinct aspects of the search results. RFP counts the fraction of negatives that appear above the lowest ranked positive (Jaakkola *et al.* 1999). Median RFP counts the fraction of negatives above the median positive. ROC_{50} gives an average over different tradeoffs between positives and negatives by integrating under the curve obtained by plotting the fraction of true positives as a function of the fraction of false positives up to the fiftieth false positive (Gribskov and Robinson 1996). $E() < 0.02$ measures the number of errors at an E-value threshold of 0.02. Finally, E-number is the number of errors at the point where the number of false positives equals the number of false negatives.

We show the classification accuracy of each of the p-value estimation methods relative to method MLH in Table 2. The statistical significance of the comparisons depends strongly on which classification accuracy metric is used. For example, compared to the regression methods, MLH has statistically superior classification accuracy under the ROC_{50} metric. It is significantly better than method REGRESS3 under the $E() < 0.02$ and E-number metrics as well. Under all other metrics, MLH is at best marginally superior to the regression methods. It is significantly better than method ALTSCHUL-GISH under all accuracy metrics. Compared to the other maximum likelihood based methods, ML and EM, classification is significantly worse using MLH under the RFP metrics but equivalent using the other metrics. This, however, is an artifact of the interaction of the distribution of target sequence lengths and the tendency of ML

and EM to overestimate the p-values of short sequences (see below).

Relationship between classification and statistical accuracy

The differences in *classification* accuracy between MLH and the other methods of p-value estimation are strongly related to differences in the *statistical* accuracy of the methods and the distribution of sequence lengths in the target database (SCOP). This can be seen by examining the plots in Fig. 5, which were compiled from the same data as Table 1 and Fig. 4. These plots show how, compared with two other methods, the classification accuracy and statistical accuracy of method MLH p-values vary as a function of target sequence length. For example, the histogram in the left panel shows that, for families whose sequences are short, ML p-values have lower classification accuracy than those of method MLH, and higher classification accuracy for longer families whose sequences are long. The curve superimposed on the histogram, the "delta PSE" curve, in that plot shows that method MLH has smaller p-value slope error for short sequences and larger p-value slope error for long sequences than method ML.

In both plots in Fig. 5, as well as in similar plots comparing different estimation methods or measuring classification accuracy with the other metrics (data not shown), there is a striking agreement in the shape of the histogram and the corresponding delta PSE curve. The explanation for this has two parts. Firstly, a rising (falling) delta PSE curve in Fig. 5 indicates that the other method "favors" (in a classification sense) longer (shorter) sequences. To see this, note that the curve in the left plot is the difference between the curves for MLH and ML in the SCOP plot in Fig. 4. Clearly, method ML strongly overestimates the p-values of short sequences, whereas the p-values estimated by method MLH are relatively accurate for all target lengths. Secondly, most SCOP families contain sequences of highly uniform length (Fig. 6). Thus, sequences of lengths far from the average length for a family are likely to be unrelated to it. A method that favors long sequences (such as method ML) will, therefore, tend to have higher classification accuracy for families with long average length than one that accurately estimates p-values. (With length-biased p-values such as those estimated by method ML, the p-values of short sequences, which would be false positives, are pushed lower in the sorted list, resulting in higher classification accuracy.) The strong correspondence in the shapes of the histogram and delta PSE curve in the right plot in Fig. 5 can be similarly explained.

If we combine the above analysis of Fig. 5 with the fact that the average sequence length of most SCOP families is above 75 (Fig. 7), we reach the conclusion that the lower classification accuracy of method MLH compared with methods ML and EM is artifactual. For example, method ML achieves higher classification accuracy (using the RFP or Median RFP) than MLH (Table 2) because it systematically overestimates the p-values of short sequences. The result is higher accuracy for ML on families with average sequence length greater than about 75, and lower accuracy on families with longer average sequence length. Since most SCOP families have average target sequence lengths greater than 75, as can be seen in Fig. 7, this classification test favors method ML.

Summary

Accurate estimates of the statistical significance of gapped local-alignment scores are important for at least two reasons. Firstly, inaccurate p-values give inaccurate estimates of the number of false positives in a database search. Secondly, inaccurate p-values lead to incorrect inferences concerning the relatedness of sequences.

The present work makes several contributions to the field of estimating the statistical distribu-

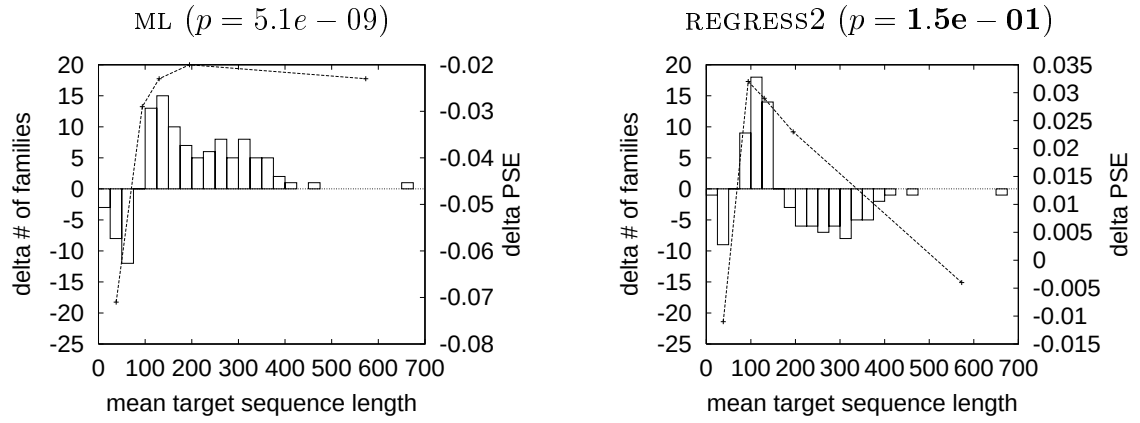

Figure 5: **Differences in classification accuracy (RFP) and p-value slope error between method MLH and other methods.** The difference in classification accuracy in the SCOP database tests is shown as a histogram of the net number of sequence families (of given average sequence length) where the named method has better accuracy. The difference in p-value slope error (MLH PSE minus other method PSE) is shown as a curve. The p-values above the plots are for the *sign test* test, with bold font indicating that method MLH had superior classification accuracy overall.

target length	50	100	250	500	1000
number of scores reported	14	32	123	298	507

Table 3: **BLAST preferentially reports scores of long, unrelated sequences.** The table shows the numbers of scores reported by BLAST in a search of a database of equal numbers (10,000) of pseudo-random protein sequences of five distinct lengths.

tions of gapped local-alignment scores. Firstly, we describe a maximum likelihood-based algorithm, MLH, that uses estimation of the entropy parameter H used in the edge effect correction to improve the accuracy of the fit to the empirical data. We show that this results in more accurate p-values across a wide range of target sequence lengths. Secondly, we show that p-value accuracy can be further increased, when there is sufficient data, by fitting a small number of extreme value distributions to the scores for target sequences in different length ranges. We call this approach "length stratification". Thirdly, we study using a two-component mixture to separately model the scores of the related and unrelated sequences in a database search. The expectation maximization algorithm is used to maximize the likelihood of this model. Somewhat surprisingly, this did not noticably improve the accuracy of the distribution estimate when compared with simply removing scores with low E-values. Fourthly, we introduce a new metric for evaluating the statistical accuracy of estimated p-values, and use it to compare maximum likelihood methods with regression-based and lookup table methods. Fifth, we show that differences in the classification power of p-values estimated by different methods are generally artifacts of the interaction of length-dependent inaccuracies in the p-values and the length distributions of sequence databases.

Discussion

This work underscores the fact that popular sequence similarity search algorithms give p-values whose accuracy depends on the length of the

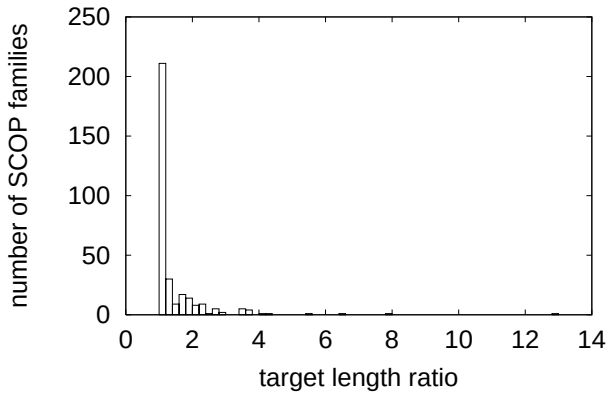

Figure 6: **Ratio of lengths of longest to shortest sequences in individual SCOP families.**

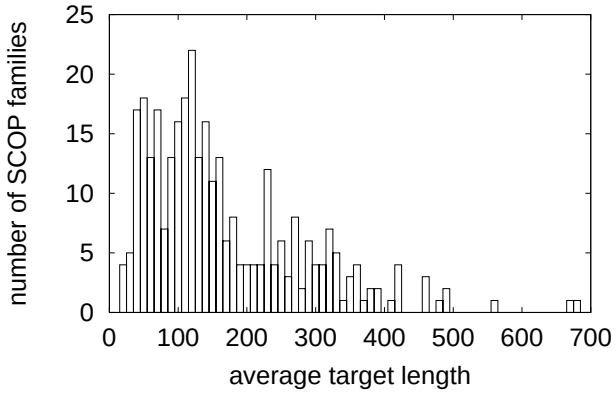

Figure 7: Average length of domain sequences in SCOP families.

target sequence. For example, an estimation method similar to that used by BLAST underestimates p-values for short sequences but gives more accurate estimates for long sequences. In contrast, the regression-based empirical method used by FASTA gives p-values that are too large for short sequences, and too small for most other lengths. The maximum likelihood method (MLH) introduced here gives more accurate p-values form almost all target sequence lengths.

We also show here that the practice of testing similarity search methods on families of protein sequences can be misleading. In particular, methods that give length-biased p-values will sometimes have higher classification accuracy than methods that report more accurate p-values. Combining plots of relative classification accuracy and relative error in p-values (comparing two p-value estimation methods) can uncover such artifacts. We argue that if sequence length is to be included as a factor in evaluating a match score, it is desirable that it be an *explicit* parameter in the p-value calculation, rather than a chance by-product of length-dependent inaccuracies in calculating p-values. A cost function argument can also be made that negative classification errors due to sequence length are more expensive than negative errors due to simple low similarity because sequence fragments are common and important in databases.

Our results also show that the many distinct classification accuracy metrics that have been used in evaluating similarity search methods can give very different results. Generally speaking, the E-number and $E() < 0.02$ metrics are rather insensitive to differences between methods compared to RFP methods. There are cases, however, when the reverse is true. The ROC_{50} methods seems to lie somewhere in between in terms of sensitivity. These differences in sensitivity are due to the different emphases of the metrics. For instance, RFP is very sensitive to the score of the lowest-scoring homolog. Since the lowest is always an outlier with high variance, this makes it an unstable metric. On the other hand, the score of the lowest-scoring homolog is of little consequence to E-number and $E() < 0.02$ which are mainly influenced by the scores of the top-scoring homologs. One should therefore use a range of classification accuracy metrics in evaluating search methods unless one is confident that all users of the methods have the same "cost function" for errors.

Previous work (Pearson 1998) showed that the empirical approach to p-value estimation should work as well for DNA as for protein sequences. The results reported here should, therefore, be

equally applicable to DNA searches despite the fact that this study focused only on protein searches.

The tendency of method ALTSCHUL-GISH to greatly underestimate the p-values of short target sequences leads naturally to the question of whether empirical p-value estimation could be used with the BLAST program. Unfortunately, empirical p-value estimation requires scores for either all target sequences, a random subset of target sequences or a score-censored subset (i.e., all sequences with scores above some fixed value). To preserve the speed of BLAST p-values must be estimated from a small subset of sequences in the database. Unfortunately, due to its heuristics, BLAST tends to compute the scores of long sequences more often than those of short sequences, making empirical fitting impractical.

To illustrate this, we ran BLAST using a single protein query against a database of pseudo-random (synthetic) sequences. The database contains 50,000 sequences, 10,000 each of lengths 50, 100, 250, 500, and 1000. As shown in Table 3, over half of the 975 scores reported by BLAST are for the longest sequences. Because of this bias, none of the methods for empirical p-value estimation discussed here would give accurate estimates from BLAST output. This could be overcome if the BLAST algorithm were modified to compute the alignment scores of a random set of sequences in the database in addition to those of sequences dictated by the BLAST heuristics. The scores of the random sequences could then be used to empirically estimate p-values for the other sequences, as is done by the current version of the FASTA algorithm (v3.1).

Availability

The algorithms described in this paper are implemented in Perl. The software is available by ftp from:
`ftp://ftp.sdsc.edu/pub/sdsc/biology/evd`.

Appendix

Expectation Maximization Method (EM)

Our maximum likelihood estimation algorithm (MLH) removes scores for all sequences whose estimated E-values are less than 1.0 and then repeats the maximum likelihood estimation. This runs the risk of not removing distant homologs whose E-values are slightly greater than 1.0. It may also mistakenly remove non-homologs with E-values less than 1.0, although, on average, it should only remove one non-homolog.

Another way to approach the problem is to consider the scores to be a mixture of two distributions. One distribution, $f_1(x|\theta_1)$, is the score distribution for non-homologs. The other distribution, $f_2(x|\theta_2)$ is that of the homologs of the query that occur in the target database. If we consider selecting target sequences from the database at random, then their scores will have the probability density function (likelihood function)

$$f(x|\theta_1,\theta_2,c) \;=\; cf_1(x|\theta_1) + (1-c)f_2(x|\theta_2),$$

where $0 \le c \le 1$ is the fraction of non-homologs in the database.

We assume that the score density function for non-homologs is the same as for method ML. For homologous sequences, we assume a Gaussian score distribution.[6] Finally, we assume a prior distribution on c that is uniform between zero and one. These assumptions can be written as

$$\begin{aligned} f_1(x_i|\theta_1) &= f_1(x_i|K,\lambda) \\ &= \lambda K N_i \exp(-\lambda x_i - K N_i e^{-\lambda x_i}), \\ f_2(x_i|\theta_2) &= f_2(x_i|\mu,\sigma) \\ &= \frac{1}{\sqrt{2\pi}\sigma}\exp(\frac{-(x_i-\mu)^2}{2\sigma^2}), \text{ and} \\ f(c) &= 1/c, \text{ for } 0 \le c \le 1. \end{aligned}$$

[6] The assumption of a Gaussian distribution for the scores of related sequences is somewhat arbitrary. This choice was made mainly because these scores probably do not follow the same distribution in all cases. Their distribution tends to be unimodal , so a Gaussian distribution seems like a reasonable compromise.

function EM(X: *list of scores*, T: *list of lengths*, q: *length of query*)
Find the values of K and λ that maximize the two-component likelihood function given the sequence scores and lengths.
Get initial estimates for $\hat{K}$, $\hat{\lambda}$ and $\hat{H}$ using **function** MLH.
Set $\hat{\mu}$ and $\hat{\sigma}$ to mean and standard deviation of outliers.
do
E-Step: Compute the value of z_i for each target sequence s_i.
M-Step: Maximize the two-component log likelihood function.
until the fractional change in the log likelihood function is less than 10^{-6}.
end

Figure 8: Expectation Maximization Algorithm

The expectation maximization (EM) procedure allows us to determine the values of θ_1, θ_2 and c that maximize the likelihood function $f(x|\theta_1, \theta_2, c)$. The procedure is given initial estimates for each of the parameters. It then refines the estimates by alternately repeating two steps, the E-step and the M-step, until convergence. The E-step (expectation step) computes the expected value of one auxiliary variable, z_i, for each sequence, given current estimates at the parameters of the model.[7] Letting $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \hat{c})$, the equation for this expected value is

$$\begin{aligned} \bar{z}_i &= E_{\hat{\theta}, x_i}[z_i] \\ &= \frac{\hat{c} f_1(x_i|\hat{\theta}_2)}{\hat{c} f_1(x_i|\hat{\theta}_1) + (1-\hat{c}) f_2(x_i|\hat{\theta}_2)}, \end{aligned}$$

where $\hat{\theta}$, $\hat{\theta}_1$, $\hat{\theta}_2$ and $\hat{c}$ are the current estimates for the parameters. The M-step (maximization step) maximizes the expected value of the augmented log likelihood function, $L(x, z|\theta_1, \theta_2, c)$. Letting $\theta = (\theta_1, \theta_2, c)$, this expectation is

$$\begin{aligned} E_{\theta', x_i}[L(x, z|\theta)] &= \sum_{i=1}^{n} [\bar{z}_i \log f_1(x_i|\theta_1) + \\ &\quad (1-\bar{z}_i) \log f_2(x_i|\theta_2) + \\ &\quad \bar{z}_i \log(c) + \\ &\quad (1-\bar{z}_i) \log(1-c)]. \quad (10) \end{aligned}$$

The expectation function (Eqn. 10) can be maximized separately over the three free parameters θ_1, θ_2 and c since it can be written as a sum of three summations each involving only one of the parameters. We maximize over c by setting the new estimate for c to

$$\hat{c} = \frac{1}{n} \sum_{i=1}^{n} \bar{z}_i.$$

To solve for $\hat{\theta}_1$ we use the Newton-Raphson algorithm (as in method ML), but the formulas for $g(\hat{\lambda})$, $g'(\hat{\lambda})$ and $\hat{K}$ become

$$\begin{aligned} g(\hat{\lambda}) &= \frac{1}{\hat{\lambda}} - \frac{\sum_{i=1}^{n} \bar{z}_i x_i}{\sum_{i=1}^{n} \bar{z}_i} + \frac{\sum_{i=1}^{n} \bar{z}_i N_i x_i e^{-\hat{\lambda} x_i}}{\sum_{i=1}^{n} \bar{z}_i N_i e^{-\hat{\lambda} x_i}}, \\ g'(\hat{\lambda}) &= -\hat{\lambda}^{-2} - \frac{\sum_{i=1}^{n} \bar{z}_i N_i x_i^2 e^{-\hat{\lambda} x_i}}{\sum_{i=1}^{n} \bar{z}_i N_i e^{-\hat{\lambda} x_i}} + \\ &\quad \left(\frac{\sum_{i=1}^{n} \bar{z}_i N_i x_i e^{-\hat{\lambda} x_i}}{\sum_{i=1}^{n} \bar{z}_i N_i e^{-\hat{\lambda} x_i}} \right)^2, \text{ and} \\ \hat{K} &= \frac{\sum_{i=1}^{n} \bar{z}_i}{\sum_{i=1}^{n} \bar{z}_i N_i e^{-\hat{\lambda} x_i}}. \end{aligned}$$

Finally, the new estimate for $\hat{\theta}_2$ is given by the equations

$$\hat{\mu} = \frac{\sum_{i=1}^{n} \bar{z}_i x_i}{\sum_{i=1}^{n} \bar{z}_i}, \text{ and}$$

[7] The variable z_i is called the "missing information" in the literature on the EM algorithm. It has the value zero if the ith sequence is not related to the query (so x_i is governed by θ_1), and one if the ith sequence is related to the query (so x_i is governed by θ_2).

$$\hat{\sigma} \;=\; \sqrt{\frac{\sum_{i=1}^{n} \bar{z}_i (x_i - \hat{\mu})^2}{\sum_{i=1}^{n} \bar{z}_i}}.$$

The complete EM method (Fig. 8) for estimating p-values begins by using **function** MLH (Fig. 1) to get initial estimates for $\hat{K}$ and $\hat{\lambda}$. The outliers that remain after the last iteration of the function (scores with E-values below 1.0) are used to get initial estimates for $\hat{\mu}$ and $\hat{\sigma}$, by setting them to the sample mean and variance, respectively, of the outlier scores. The initial estimate for $\hat{c}$ is set to the number of non-outliers divided by the total number of scores. Then, the E- and M-steps are applied alternately to refine the estimates of all the parameters. This stops when the fractional change in the log likelihood is less than 10^{-6}.

*Previously published methods (*REGRESS1*,* REGRESS2*,* REGRESS3*, and* ALTSCHUL-GISH*)*

We compare our new methods with the three regression based methods and one method that uses pre-calculated parameters for the extreme value distribution (method ALTSCHUL-GISH), all of which are described in Pearson (1998). We used Pearson's implementation of these methods as contained in file `scaleswn.c` which is part of the FASTA package. We obtained it by downloading file `ftp://ftp.virginia.edu/pub/fasta/fasta33t01.shar`. We wrote a main routine that reads in a list of scores and sequence lengths and passes them to the appropriate routines in `scaleswn.c` for processing and conversion to p-values.

The regression based methods assume that the average local alignment score increases linearly with the logarithm of the length of the target sequence. One method (REGRESS3) introduces the further assumption that the score variance also varies linearly with the logarithm of the length of the target sequence. The other two Pearson methods (REGRESS1 and REGRESS2) assume that the score variance is independent of target sequence length. Although not mentioned in Pearson (1998), these two methods can be shown to be equivalent to assuming the score distribution

$$P(S \geq x) \;=\; 1 - \exp(-K(qt)^u e^{-\lambda x}),$$

where u is an additional parameter that determines how average score varies with target sequence length.

Pearson's three regression-based methods estimate the mean, $\mu(t)$, and variance, $\sigma(t)$ of the score distribution from scores of (putatively) unrelated target sequences. (Recall that t is the length of the target sequence.) All three methods assume that $\mu(t)$ is a linear function of $\log(t)$. Methods REGRESS1 and REGRESS2 assume $\sigma(t)$ is constant whereas REGRESS3 assumes it is linear in $\log(t)$. The methods then convert raw scores to Z-scores using the equation

$$Z(S) \;=\; \frac{S - \mu(t)}{\sigma(t)}.$$

Finally, they estimate p-values using the formula

$$p(Z) \;=\; 1 - \exp(-e^{-Z\pi/\sqrt{6} - \Gamma'(1)}),$$

where $\Gamma'(1)$ is the first derivative of the gamma function evaluated at 1 (approximately 0.577216).

Besides differing in their assumptions about the dependence of score average and variance, Pearson's methods also differ in how the scores of sequences related to the query are eliminated during estimation of $\mu(t)$ and $\sigma(t)$. In all three methods, scores are placed into bins segregated according to the length of the target sequence. REGRESS1 and REGRESS2 compute $\mu(t)$ using linear regression on the means of the bins. Scores with very high or low Z-values are removed from the bins and regression is repeated. Bins with large variance are then removed and regression is repeated. REGRESS3 additionally performs a regression on the standard deviations of the bins to get $\sigma(t)$ whereas

REGRESS1 uses the average standard deviation of the bins as $\sigma(t)$ for all t. The other method, REGRESS2, uses regression on the means of the bins, excludes bins with high and low residual variance, repeats the regression, then removes sequences with very high or very low Z-values. These steps are repeated up to five times.

The pre-computed parameters of method ALTSCHUL-GISH were calculated from random simulations by Altschul and Gish (1996) using a fixed sequence model for the query and target sequences. This method assumes that local alignment scores for target sequences of length t follow the same extreme value distribution as our MLH method, and uses those pre-computed parameters.

References

Stephen F. Altschul and Warren Gish. Local alignment statistics. *Methods in Enzymology*, 266:460–480, 1996.

Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinhui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. Gapped-BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

Amos Bairoch and R. Apweiler. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Research*, 28:45–48, 1994.

Steven E. Brenner, Cyrus Chothia, and Tim Hubbard. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proceedings of the National Academy of Sciences USA*, 95:6073–6078, 1998.

Christofer Chatfield. *Statistics for Technology.* Chapman and Hall, 1983. Third Edition.

Sean R. Eddy. Multiple alignment using hidden Markov models. In C. Rawlings *et al.*, editor, *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 114–120, Menlo Park, California, 1995. AAAI Press.

Michael Gribskov and Nina L. Robinson. The use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Computers and Chemistry*, 20:25–33, 1996.

E. J. Gumbel. *Statistics of extremes.* Columbia University Press, 1958.

Tommi Jaakkola, Mark Diekhans, and David Haussler. Using the fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 149–158, Menlo Park, California, 1999. AAAI Press.

Samuel Karlin and Stephen F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences USA*, 87:2264–2268, 1990.

J. F. Lawless. *Statistical Models and Methods for Lifetime Data.* John Wiley & Sons, Inc., New York, 1982.

Richard Mott and Roger Tribe. Approximate statistics of gapped alignments. *Journal of Computational Biology*, 6:91–112, 1999.

Richard Mott. Maximum-likelihood estimation of statistical distributions of Smith-Waterman local sequence similarity scores. *Bull. Math. Biol.*, 54:59–75, 1992.

William R. Pearson. Empirical statistical estimates for sequence similarity searches. *Journal of Molecular Biology*, 276:71–84, 1998.

T. J. Santner and D. E. Duffy. *The Statistical Analysis of Discrete Data.* Springer-Verlag, 1989.

Temple Smith and Michael Waterman. Identification of common molecular subsequences.

Journal of Molecular Biology, 147:195–197, 1981.

Rainer Spang and Martin Vingron. Statistics of large-scale sequence searching. *Bioinformatics*, 14:279–284, 1998.